A method of hp-adaptation for Residual Distribution schemes

Quentin Viville

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Quentin Viville**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Mathématiques appliquées

Construction d'une méthode hp-adaptative pour les schémas aux Résidus Distribués

---

Directeur de thèse: Rémi Abgrall

Co-Directrices de thèse: Héloïse Beaugendre et Cécile Dobrzynski

Date de soutenance: 22 novembre 2016

Après avis des rapporteurs:

Frédéric Hecht ..	Professeur, Université Pierre et Marie Curie ....
Boniface Nkonga	Professeur, Université de Nice - Sophia Antipolis

Devant la commission d'examen composée de:

Rémi Abgrall	Professeur, Universität Zürich .....
Héloïse Beaugendre	Maître de Conférences, INP Bordeaux .....
Cécile Dobrzynski ..	Maître de Conférences, INP Bordeaux .....
Frédéric Hecht	Professeur, Université Pierre et Marie Curie .....
Adam Larat	Chargé de Recherche CNRS, École Centrale Paris
Boniface Nkonga ...	Professeur, Université de Nice - Sophia Antipolis .

– 2016 –

Résumé Cette thèse présente la construction d'un schéma aux Résidus Distribués p-adaptatif pour la discrétisation des équations d'Euler ainsi qu'un schéma aux Résidus Distribués hp-adaptatif pour les équations de Navier-Stokes pénalisées. On rappelle tout d'abord les équations d'Euler et de Navier-Stokes ainsi que leurs versions non dimensionnelles. Les définitions et propriétés de base des schémas aux Résidus Distribués sont ensuite présentées. On décrit alors la construction d'un schéma aux Résidus Distribués p-adaptatif pour les équations d'Euler. La construction du schéma p-adaptatif est basée sur la possibilité d'exprimer le résidu total d'un élément K de degré k (au sens où l'élément fini (K, P, Σ) est un élément fini de degré k) comme une somme pondérée des résidus totaux de ses sous-éléments de degré 1. La solution discrète ainsi obtenue est en général discontinue à l'interface entre un élément subdivisé et un élément non subdivisé. Ceci contredit l'hypothèse de continuité de la solution qui est utilisée pour démontrer le théorème de Lax-Wendroff discret pour les schémas aux Résidus Distribués. Cependant, on montre que cette hypothèse peut être assouplie. La conséquence pratique est que si l'on emploie des quadratures particulières dans l'implémentation numérique, on peut quand même démontrer le théorème de Lax-Wendroff discret, ce qui garantit la convergence du schéma numérique vers une solution faible des équations d'origine. Les formules qui permettent d'exprimer le résidu total comme une somme pondérée des résidus totaux des sous-éléments sont à la base de la méthode de p-adaptation présentée ici. Dans le cas quadratique, la formule est obtenue avec les classiques fonctions de base de Lagrange en dimension deux et avec des fonctions de base de Bézier en dimension trois. Ces deux formules sont ensuite généralisées à des degrés polynomiaux quelconques en dimension deux et trois avec des fonctions de base de Bézier. Dans la deuxième partie de la thèse, on présente l'application du schéma p-adaptatif aux équations pénalisées de Navier-Stokes avec adaptation de maillage anisotrope. En pratique, on combine le schéma p-adaptatif avec la méthode IBM-LS-AUM (Immersed Boundary Method with Level Sets and Adapted Unstructured Meshes). La méthode IBM-LS-AUM permet d'imposer les conditions aux bords grâce à la méthode de pénalisation et l'adaptation anisotrope du maillage à la solution numérique et à la level-set augmente la précision de la solution et de la représentation de la surface. Une fois la méthode IBM-LS-AUM combinée avec le schéma p-adaptatif, il est alors possible d'utiliser des éléments d'ordre élevés en-dehors de la zone où la pénalisation est appliquée. La méthode est robuste comme le montrent les diverses expérimentations numériques à des vitesses faibles à élevées et à différents nombres de Reynolds.

Title A method of hp-adaptation for Residual Distribution schemes

Abstract This thesis presents the construction of a p-adaptive Residual Distribution scheme for the steady Euler equations and a hp-adaptive Residual

Distribution scheme for the steady penalized Navier-Stokes equations in dimension two and three. The Euler and Navier-Stokes equations are recalled along with their non dimensional versions. The basis definitions and properties of the steady Residual Distribution schemes are presented. Then, the construction of a p-adaptive Residual Distribution scheme for the Euler equations is considered. The construction of the p-adaptive scheme is based upon the expression of the total residual of an element of a given degree k (in the Finite Element sense) into the total residuals of its linear sub-elements. The discrete solution obtained with the p-adaptive scheme is then a one degree polynomial in the divided elements and a k-th degree polynomial in the undivided ones. Therefore, the discrete solution is in general discontinuous at the interface between a divided element and an undivided one. This is in apparent contradiction with the continuity assumption used in general to demonstrate the discrete Lax-Wendroff theorem for Residual Distribution schemes. However, as we show in this work, this constrain can be relaxed. The consequence is that if special quadrature formulas are employed in the numerical implementation, the discrete Lax-Wendroff theorem can still be proved, which guaranties the convergence of the p-adaptive scheme to a weak solution of the governing equations. The formulas that express the total residual into the combination of the total residuals of the sub-elements are central to the method. In dimension two, the formula is obtained with the classical Lagrange basis in the quadratic case and with the Bézier basis in dimension three. These two formulas are then generalized to arbitrary polynomial degrees in dimension two and three with a Bézier basis. In the second part of the thesis the application of the p-adaptive scheme to the penalized Navier-Stokes equations with anisotropic mesh adaptation is presented. In practice, the p-adaptive scheme is used with the IBM-LS-AUM (Immersed Boundary Method with Level Sets and Adapted Unstructured Meshes) method. The IBM-LS-AUM allows to impose the boundary conditions with the penalization method and the mesh adaptation to the solution and to the level-set increases the accuracy of the representation of the surface and the solution around walls. When the IBM-LS-AUM is combined with the p-adaptive scheme, it is possible to use high-order elements outside the zone where the penalization is applied. The method is robust as shown by the numerical applications at low to large Mach numbers and at different Reynolds in dimension two and three.

Keywords Residual Distribution schemes, High-order methods, p-adaptation, hp-adaptation, Anisotropic mesh adaptation, Euler equations, Navier-Stokes equations, Compressible flows

Mots-clés Schémas aux Résidus Distribués, Méthodes d'ordre élevé, p-adaptation, hp-adaptation, Adaptation de maillage anisotrope, Équations d'Euler, Équations de Navier-Stokes, Écoulements compressibles

Laboratoire d'accueil Inria Bordeaux (Équipe Cardamom). Institut de Mathématiques de Bordeaux.

Remerciements

Je remercie mon directeur de thèse, M. Rémi Abgrall, ainsi que mes co-directrices de thèse, Mme. Héloïse Beaugendre et Mme. Cécile Dobrzynski, d'avoir accepté d'encadrer ces travaux. Je les remercie pour m'avoir guidé tout au long de ces trois années, pour avoir répondu sans relâche à mes innombrables questions, tout en me laissant une certaine autonomie afin que je puisse explorer de nouvelles pistes. En particulier, je remercie Héloïse pour son aide, ses conseils et la gentillesse de son accueil lors de mes très nombreuses visites à son bureau.
Je remercie M. Frédéric Hecht et M. Boniface Nkonga d'avoir accepté de rapporter ma thèse et de faire partie du jury. Je remercie aussi M. Adam Larat d'avoir accepté de faire partie du jury.
Enfin, je salue tous mes amis de l'Inria et bien sûr, je remercie ma famille pour leur soutien.

Contents

Chapter 1

Introduction

1.1 Continuous and discrete problem

Many physical, biological, mechanical or economical phenomena are modeled with partial differential equations and progress in these sciences requires the study of the solutions to these equations. The analytical approach consists generally in proving existence, uniqueness and finding a representation of the exact solution to these equations. Unfortunately, a lot of partial differential equations are difficult to solve analytically if not impossible, especially for non linear equations. One famous example is the prize of the millennium for the Navier-Stokes existence and smoothness problem. Thus, the alternative comes from numerical methods which provide tools to solve approximately partial differential equations. In many industries and research fields such as aeronautics, petroleum, nuclear, biology, medical, weather forecast and several others, dramatic technological progress have been obtained by using numerical methods to study quantitatively the properties of discrete solutions to partial differential equations. In particular, the application of numerical methods to the problems of fluid dynamics has given birth to a new field of research called "computational fluid dynamics" widely known as "CFD". If we look in particular at the conception of aircrafts, until the end of the 1960's fluid dynamics problems were approached with theory and experiments [20] as pure theory was used to design planes and then real experiments were used to test their flying abilities within ground test facilities like wind tunnels. This approach has two main limits. The first one is that as said above the theoretical study of the partial differential equations governing fluid dynamics is complicated and sometimes hopeless. The second one is that practical experiments is not a feasible approach in many cases. For example, imagine the design of a hypersonic trans-atmospheric vehicle, the construction of a wind tunnel simulating the high speed flow and the extreme temperature would be, to say the least, impractical. CFD brings a practical solution to both these problems and complements pure theory and practical experiments.

In the first part of the section, we present a review of the current numerical methods used for research in CFD and we invoke the growing interest for high order methods and adaptations techniques. In the second part of the section we present the immersed boundary methods which are a class of methods used to treat obstacles in CFD simulations.

1.2 Numerical methods

Today, most industrial CFD codes are based on second-order Finite Volume methods. When dealing with complex geometries discretized with unstructured irregular highly stretched meshes, the order of these methods degrade generally to an order between one and two. From the conclusions of the 1st International Workshop on High-Order CFD Methods [88], Finite Volume methods produce results with insufficient accuracy with the typical grid size used for industrial applications. Strong mesh refinement in order to produce sufficiently accurate solutions is not a practical solution as it leads to long computation times. A possible solution is to use high order methods as they theoretically offer a greater accuracy than low-order methods, or more precisely, the computational cost to achieve a given error level is lower with a high-order method than with a low order one. In this sense, high order method are more accurate for the same computational time, or equivalently they are faster for the same accuracy level. However, high order methods, in spite of their potential, bring several challenges. They are inherently more difficult to design and implement than low order methods, and require more memory. They offer much less numerical diffusion; this can be seen as a good thing as it brings higher fidelity with respect to the original equation but on the other hand the approximation of discontinuities can cause difficulties for the numerical code. In this sense, the ideal approach would be to combine a high order method with hp-adaptation, where the order is degraded and the mesh size is reduced in the vicinity of shocks.

1.2.1 A review of the principal currently researched and developed high order methods and adaptation techniques for CFD.

We are presenting now four high order methods that are currently researched and developed in CFD.

Finite Volumes / ENO / WENO

The Finite Volume method is based on a control volume formulation. The mesh of the domain is made of elements called control volumes and the gov-

erning equations are integrated over each control volume such that the discrete formulation expresses the conservation principle of the quantities studied. In multiple dimensions, most finite volume schemes suffer a reduction of accuracy with irregular meshes as the one-dimensional formulation is applied along particular mesh directions. The extension to higher orders, resulting in the so called ENO and WENO schemes [80], are based upon the idea of adaptive stencils in the reconstruction procedure based on local smoothness of the numerical solution. They require very large stencils for unstructured meshes and consequently it might be difficult with these schemes to implement an efficient parallelization and high order grid adaptation.

Finite Element Method

The Finite Element method is based on a variational formulation of the conservation equation, which allows to define a space that contains the solution of the weak form of the equation. Then an approximate solution is built in a finite dimension approximate space of the original space. The mesh of the domain is divided into elements and the discrete solution is written on each element as a linear combination of basis functions that have a compact support and the discrete solution is required to be continuous at the interface between adjacent elements. This construction gives a compact stencil like in Discontinuous Galerkin methods but with a lower number of degrees of freedom. Usually, the space that contains the test functions is the same as the space that contains the discrete solution and the resulting method is called the "Galerkin method". In the cases where advection dominates diffusion, this approach can bring spurious oscillations [39] that render the scheme unstable. The usual solution in CFD to this problem is to add a dissipation term. Several stabilization techniques have been proposed [57]. One in particular that has attracted a lot of attention in the CFD community is the so-called SUPG method [27, 58, 46, 62, 57, 42, 43, 61], where a stabilization term is added to the Galerkin discretization to compensate for the absence of dissipation in the direction of the flow. In addition, the diffusive part of the scheme is augmented with a numerical diffusion term in order to stabilize the discrete solution in the vicinity of shocks. Finally, the Finite Element method can be combined with hp-adaptation while the discrete solution remains globally continuous, as an example an implementation of a SUPG hp-adaptive Finite Element scheme for the compressible Reynolds Averaged Navier-Stokes equations with a Spalart-Allmaras turbulence model is proposed in [19].

Discontinuous Galerkin Methods

The discontinuous Galerkin method is as its names implies based on a finite element formulation but the discrete solution can be discontinuous at the interface of adjacent elements. The discrete solution is a polynomial on each

element, as in FEM, but the discontinuity between two adjacent elements is handled by numerical fluxes as in Finite Volume methods. As such, it is considered as an hybrid method of FEM and Finite Volume methods. DG methods allows theoretically to construct a discrete solution of arbitrary high order with a very compact stencil as the solution in each element is by design independent of the solution in other elements.This leads to a compact scheme well suited for hp-adaptation, diverse type of elements and mesh topologies and for efficient parallelization. The use of high order polynomials produces oscillations in the presence of shocks that are resolved by techniques like artificial viscosity [53] or local projection methods [28, 31]. Finally, the main drawback is the computational cost. Indeed, the discontinuous nature of the scheme implies that it uses more degrees of freedom than continuous methods like FEM or Residual Distribution schemes [10, 17].

Residual Distribution Schemes

The Residual Distribution (RD) schemes are the schemes used in this thesis. Like in the Finite Element method, a Residual Distribution scheme uses a continuous approximation of the solution. It is based on an integral formulation of the equations of conservation written for each element of the mesh. This allows to define for each element a total residual that is distributed to the degrees of freedom of the element, following a strict distribution rule. The main feature of the scheme is that it is designed to achieve a maximum principle without any parameter to tune [5]. Theoretical and practical results have shown that the discrete solution is a high order approximation of the solution of the governing equations for inviscid and viscous flows in the steady case. The compact nature of the scheme makes it compatible with parallelization. However, p-adaptation for Residual Distribution schemes is not evident because of the continuity constrain of the discrete solution.
A discontinuous Residual Distribution scheme has been proposed in [4] for hyperbolic equations. The discrete solution is therefore not necessarily continuous at the interface of two adjacent elements and non conformal elements can be used, which renders the scheme comparable to a Discontinuous Galerkin scheme and presumably well suited for hp-adaptation. However, like Discontinuous Galerkin schemes, the computational cost of the discontinuous Residual Distribution schemes is higher than the computational cost of the continuous Residual Distribution schemes.

Adaptation methods

As shown by the review above, high order methods in CFD have made considerable progress. However, their efficiency and so their justification is still controversial. Indeed, real life problems from physics have shown that flows are in general discontinuous and exhibit shocks. The consequence is that high

order methods are not suited in zones where discontinuities happen whereas they are more efficient in smooth regions [22, 85]. The solution is then to use hp-adaptation and as a consequence there has been a considerable amount of research on adaptation methods for CFD, in particular for the Discontinuous Galerkin method due to its highly compact nature [55, 76, 77, 84, 53, 60, 90]. Although SUPG methods have been extensively researched, it has only been recently that rigorous results for hp-adaptive techniques in the frame of SUPG methods have been established [17, 19]. Finally, for Residual Distribution schemes, to the best of our knowledge, most of the research has been done using anisotropic mesh adaptation in the particular case of the penalized Navier-Stokes equations in the frame of an immersed boundary method [24].

1.2.2 Immersed boundary methods

As described above, the mesh adaptation techniques that have been implemented so far within a Residual Distribution scheme have been mostly driven by the use of an immersed boundary method. The immersed boundary methods have become of interest to the CFD community. Indeed, the treatment of obstacles in CFD simulations is a complex problem that is handled by different techniques. Among these techniques we can cite the body fitted mesh approach [64, 44, 45] and the fictitious domain approach, also known as immersed boundary method or Cartesian method [75]. The body-fitted mesh technique seems to have retained interest in the CFD community, however when dealing with moving bodies the governing equations need to be recast in an Arbitrary Lagrangian-Eulerian frame of reference which leads to some complications like topology reconstruction or mesh interpolation [52, 65]. The class of immersed boundary methods provides an alternative to the body-fitted approach. The method was first designed for structured grids from a simple idea: the solid bodies are inside a background grid that they blank out, and the cells that are cut require a special treatment. The method has been developed for unstructured grids too [66, 83], but most developments have been made for structured grids [71, 56, 72, 21]. Even if there are now different types of immersed boundary methods that can handle incompressible, compressible viscous and turbulent flows, the treatment of wall boundary conditions remains complicated. An alternative approach presented in [8] consists in using a penalization method to enforce the boundary conditions on the surface of bodies that are described by a level-set method and embedded in an unstructured grid. It is in practice used as follows: the solids are localized with a level-set function and the mesh is adapted with respect to the level set to increase the quality of the surface representation, the wall boundary conditions are enforced with a penalization term and the mesh is again adapted to the solution obtained and to the level set. Finally, the mesh is automatically adapted to limit the increase of the number of elements, so that the computational time

remains comparable with body-fitted techniques. This method simplifies the meshing generation and the imposition of the wall boundary conditions which is usually an issue with other immersed boundary methods.

1.3 Motivation and scope of this work

With their compact nature, low computational cost and stability, the continuous Residual Distribution schemes constitute a serious alternative to the other high order schemes currently researched and developed for CFD. However, no p-adaptation mechanism has been presented so far in the frame of continuous Residual Distribution scheme. The work presented here aims to add p-adaptive capabilities to steady Residual Distribution schemes. The goal is then to construct a p-adaptive scheme that brings robustness in the sense that complex test cases with very low to very high speed flows can be run, in dimension two and three in the non viscous and viscous cases with different Reynolds numbers. Moreover, the p-adaptive scheme should be theoretically able to be extended to polynomial orders higher than two. This is not straightforward as the p-adaption mechanism relies on the expression of the total residual into a combination of total residuals on the sub-elements and the formulas found at the beginning of the thesis for polynomials of degree two seemed very complicated to generalize to arbitrary orders. Finally, the p-adaptive scheme needs to be compatible with the penalization method for the Navier-Stokes equations with the anisotropic mesh adaptation presented in [8] so that it constitutes an hp-adaptive Residual Distribution scheme for the Navier-Stokes penalized equations. However, the extension of the methods presented here to the unsteady case have not been envisaged yet. The construction of a high order Residual Distribution scheme for time dependent equations is still an area of active research and so the construction of an hp-adaptive scheme for the unsteady case constitutes a longer term goal.

1.4 Contribution of this thesis

The objective of this thesis is to build a p-adaptive Residual Distribution scheme that can be further integrated to the IBM-LS-AUM method to give a hp-adaptive penalized Residual Distribution scheme. The first step was to define the theoretical material for the construction of a p-adaptive Residual Distribution scheme. These theoretical results have been tested with the scalar two-dimensional advection and Burgers equations. The next objective was to implement the scheme in the platform RealFluid for the Euler equations in dimension two and three. Some test cases, in particular with strong shocks can now be run with the p-adaptive scheme which was not possible with the classical high order scheme. In this sense the p-adaptive scheme is more ro-

bust. This part of the work corresponds to the chapter 2, which is presented as in a paper that has been submitted and is under review. The next step was to test the scheme with the Navier-Stokes equations in dimension two and three at various speed regimes. Finally, the last objective of the thesis was to combine the p-adaptive scheme with the IBM-LS-AUM algorithm in order to obtain a penalized hp-adaptive Residual Distribution scheme. This part of the work corresponds to the chapter 3, which is presented as in the second paper which has been submitted.
We are using here the terms p-adaptation and hp-adaptation. In the literature, in particular in the context of Discontinuous Galerkin methods and Finite Element methods, the term p-adaptation describes a local change of the polynomial degree of the shape functions of the element and the term h-adaptation describes the division of the element into smaller elements of the same degree or the inverse operation. In the work presented here, the term p-adaptation denotes in the case of de-refinement that an element of a given polynomial degree k is split into smaller elements with a polynomial degree equal to one and in the case of refinement it describes the inverse operation. The term h-adaptation denotes here a classical h-adaptation technique in the sense that elements can be split or regrouped, but they can be stretched too, so more precisely we are using here an anisotropic mesh adaptation technique. The technique is described in chapter 3. Finally, the hp-adaptation denotes here the combination of h and p-adaptation in the sens defined above.

1.5 The equations of fluid mechanics and their discretization

1.5.1 Conservation laws

A conservation law is a physical law that expresses the fact that a given quantity, like the mass, the momentum or the energy in the case of fluid mechanics is conserved as time changes [68]. To express mathematically a conservation law, a control volume V with boundary ∂V is considered, $\mathbf{u}$ denotes the vector of conserved variables, $\mathbf{f}$ the flux vector and $\mathbf{n}$ the unit vector normal to the volume V. The conservation law is then expressed as follows

$$\frac{d}{dt}\int_V \mathbf{u}\, dx + \int_{\partial V} \mathbf{f}\cdot\mathbf{n}\, dx = 0. \tag{1.1}$$

The above equation expresses the fact that any variation of the solution in time is due and only due to a variation of the flux on the boundary of V. Now if we suppose that the flux is regular enough in the volume V, we have with the divergence theorem

$$\frac{d}{dt}\int_V \mathbf{u}\, dx + \int_V \text{div}\ \mathbf{f}\, dx = 0. \tag{1.2}$$

This equation is true for any volume V, so from the properties of the integral we have

$$\frac{\partial \mathbf{u}}{\partial t} + \text{div } \mathbf{f} = 0 \tag{1.3}$$

and the equation above is the differential form of the conservation law. In the work presented here, the focus is on the Euler and Navier-Stokes equations, which are conservation laws in the sense defined above.

1.5.2 The Euler equations

We present now the Euler equations, which are a system of non linear equations that describe the flow of a compressible fluid considered adiabatic and non viscous. In conservative form, the system writes

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathsf{v}) = 0 \\ &\frac{\partial \rho \mathsf{v}}{\partial t} + \text{div}(\rho \mathsf{v} \otimes \mathsf{v}) + \nabla p = 0 \\ &\frac{\partial E}{\partial t} + \text{div}\big((E+p)\mathsf{v}\big) = 0 \end{aligned} \tag{1.4}$$

where ρ is the density, v is the speed of the fluid, $E = \rho e^t$ is the total energy per unit mass and p is the pressure. We consider a perfect gas and the Euler equations are completed with the following relations for the energy

$$e^t = e + \frac{\mathsf{v} \cdot \mathsf{v}}{2} \quad \text{and} \quad e = \frac{1}{\gamma - 1} RT, \tag{1.5}$$

with the relation of state for a perfect gas

$$p = \rho R T \tag{1.6}$$

where $\gamma = 1.4$ is the Heat capacity ratio, T is the gas temperature, R the gas constant which is $287 N.m/kg$ at sea level.

This system of equations can be written in the vector form (1.3)

$$\frac{\partial \mathbf{u}}{\partial t} + \text{div } \mathbf{f}(\mathbf{u}) = 0 \tag{1.7}$$

where the vector $\mathbf{u}$ and the flux $\mathbf{f} = \mathbf{f}(\mathbf{u})$ are respectively defined by

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathsf{v} \\ \rho E \end{pmatrix}, \qquad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \rho \mathsf{v} \\ \rho \mathsf{v} \otimes \mathsf{v} + p \ \text{Id}_{d \times d} \\ (\rho E + p)\mathsf{v} \end{pmatrix}. \tag{1.8}$$

The vector $\mathbf{u} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^l$ is the vector of l conserved variables (we have dropped the dependence in time), Ω is an open set of $\mathbb{R}^d$, $d = 2, 3$ and $\mathbf{f} = (\mathbf{f}_1, ..., \mathbf{f}_d)$ is the flux vector where $\mathbf{f}_i(\mathbf{u}) : \Omega \to \mathbb{R}^l$, $i = 1, ..., d$.

The non dimensional Euler Equations

In order to avoid the cohabitation in the numerical code of very large and very small numbers, the variables of the equations are made non dimensional by dividing them by reference variables. This procedure gives the following non dimensional variables

$$\rho^* = \frac{\rho}{\rho_r}, p^* = \frac{p}{p_r}, E^* = \frac{E}{\mathsf{v}_r^2}, \mathsf{v}^* = \frac{\mathsf{v}}{\mathsf{v}_r}, \mathbf{x}^* = \frac{\mathbf{x}}{l_r}, t^* = \frac{t}{t_r}, \tag{1.9}$$

where the index r defines the reference variable and the index $*$ the non dimensional variable.

Setting

$$p_r = \rho_r \mathbf{v}_r^2 \tag{1.10}$$

we obtain the non dimensional Euler equations

$$\begin{aligned} &\frac{\partial \rho^*}{\partial t^*} + \nabla^* \cdot (\rho^* \mathsf{v}^*) = 0 \\ &\frac{\partial \rho^* \mathsf{v}^*}{\partial t^*} + \nabla^* \cdot (\rho^* \mathsf{v}^* \otimes \mathsf{v}^*) + \nabla p^* = 0 \\ &\frac{\partial E^*}{\partial t^*} + \nabla^* \cdot \left((E^* + p^*) \mathsf{v}^* \right) = 0 \ . \end{aligned} \tag{1.11}$$

In the nondimensionalization process presented here, we have chosen as reference variables the free stream pressure and the free stream speed of sound:

$$\rho_r = \rho_\infty, \mathbf{v}_r = c_\infty. \tag{1.12}$$

This choice is a classical choice for the numerical simulation of compressible flows [68]. Indeed, with this nondimensionalization, the free stream non dimensional speed has a norm equal to the free stream Mach number and the free stream pressure is constant:

$$\rho^* = \frac{\rho}{\rho_\infty}, p^* = \frac{p}{\rho_\infty c_\infty^2}, E^* = \frac{E}{c_\infty^2}, \mathsf{v}^* = \frac{\mathsf{v}}{c_\infty}, \mathbf{x}^* = \frac{\mathbf{x}}{l}, t^* = \frac{t}{l/c_\infty} \tag{1.13}$$

where $l = l_r$ is the reference length.

Finally, with this nondimensionalization process the free stream non dimensional variables are defined by

$$\rho_\infty^* = 1, p_\infty^* = \frac{1}{\gamma}, E_\infty^* = \frac{1}{\gamma(\gamma - 1)}, \mathsf{v}_\infty^* = \frac{\mathsf{v}_\infty}{c_\infty} = M_\infty. \tag{1.14}$$

1.5.3 The Navier-Stokes equations

The Navier-Stokes equations represent a more realistic model than the Euler equations. Indeed they take into account the effects of viscosity and heat

transfer. In conservative form, they write

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathsf{v}) = 0 \\
&\frac{\partial \rho \mathsf{v}}{\partial t} + \nabla \cdot (\rho \mathsf{v} \otimes \mathsf{v}) + \nabla p = \nabla \cdot \tau \\
&\frac{\partial E}{\partial t} + \nabla \cdot \big((\rho E + p)\mathsf{v}\big) = \nabla \cdot (\tau \cdot \mathsf{v} - \mathsf{q})
\end{aligned} \tag{1.15}$$

where ρ is the density, v the speed of the fluid, E the total specific energy, p the pressure, τ the viscous stress tensor and q is the heat flux.

The viscous stress tensor is defined with the Stoke hypothesis

$$\tau = -\frac{2}{3}\mu(\nabla \cdot \mathsf{v})\mathbb{I} + \mu(\nabla^t \mathsf{v} + \nabla \mathsf{v}) \tag{1.16}$$

where μ is the viscosity coefficient, computed with the Sutherland law of viscosity

$$\mu = \mu_0 \frac{T_0 + C}{T + C} \big(\frac{T}{T_0}\big)^{\frac{3}{2}} \tag{1.17}$$

where C, T_0 and μ_0 are constants such that

$$C = 110.5[K],\ T_0 = 273.15[K] \text{ and } \mu_0 = 1.716e^{-5}[\frac{kg}{ms}]. \tag{1.18}$$

The heat flux is computed with the Fourier law

$$\mathsf{q} = -\kappa \nabla T, \tag{1.19}$$

where κ is the thermal conductivity defined by the viscosity and the Prandtl number

$$\kappa = \frac{\gamma R \mu}{(\gamma - 1)\mathrm{Pr}} \tag{1.20}$$

with $\mathrm{Pr} = 0.72$.

This system of equations can be written as in the case of the steady Euler equations under the vector form (1.3)

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \ \mathbf{f}(\mathbf{u}) = 0 \tag{1.21}$$

where $\mathbf{u}$ is the vector of conserved variables and where $\mathbf{f}(\mathbf{u})$ is the flux function respectively defined by

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathsf{v} \\ \rho E \end{pmatrix}, \qquad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \rho \mathsf{v} \\ \rho \mathsf{v} \otimes \mathsf{v} + p\ \mathrm{Id}_{d \times d} - \tau \\ (\rho E + p)\mathsf{v} - \tau \cdot \mathsf{v} + \mathsf{q} \end{pmatrix}. \tag{1.22}$$

If the flux is split into the advective and diffusive part, then the equation (1.21) can be written as (in the steady case)

$$\nabla \cdot \ \mathbf{f}^a(\mathbf{u}) = \nabla \cdot \ \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) \tag{1.23}$$

where the advection flux and the diffusion flux are respectively defined by

$$\mathbf{f}^a(\mathbf{u}) = \begin{pmatrix} \rho \mathsf{v} \\ \rho \mathsf{v} \otimes \mathsf{v} + p \ \mathrm{Id}_{d\times d} \\ (E+p)\mathsf{v} \end{pmatrix} \quad \text{and} \quad \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{pmatrix} 0 \\ \tau \\ \tau \cdot \mathsf{v} - \mathsf{q} \end{pmatrix}. \tag{1.24}$$

The flux function being homogeneous with respect to the gradient of conservative variable it can be written as

$$\mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) = \mathbb{K}(\mathbf{u}) \nabla \mathbf{u} \tag{1.25}$$

where the tensor $\mathbb{K} \in \mathbb{R}^{p\times d\times p\times d}$ is the Jacobian matrix of the viscous flux defined by

$$\mathbb{K}(\mathbf{u}) = \frac{\partial \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u})}{\partial \nabla \mathbf{u}} \tag{1.26}$$

and for convenience we denote by $\mathbb{K}_{ij}$ the matrix

$$\mathbb{K}_{ij}(\mathbf{u}) = \frac{\partial \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u})}{\partial (\frac{\partial \mathbf{u}}{\partial x_j})} \ . \tag{1.27}$$

The non dimensional Navier-Stokes equations

The nondimensionalization process is the same as for the Euler equations, with the nondimensionalization of the temperature and viscosity:

$$T^* = \frac{T}{T_r} \text{ and } \mu^* = \frac{\mu}{\mu_r}. \tag{1.28}$$

With this nondimensionalization process and by setting $p_r = \rho_r \mathbf{v}_r^2$ as for the Euler equations, we obtain the non dimensional Navier-Stokes equations

$$\begin{aligned} &\frac{\partial \rho^*}{\partial t^*} + \nabla^* \cdot (\rho^* \mathsf{v}^*) = 0 \\ &\frac{\partial \rho^* \mathsf{v}^*}{\partial t^*} + \nabla^* \cdot (\rho^* \mathsf{v}^* \otimes \mathsf{v}^*) + \nabla p^* = \nabla^* \cdot \tau^* \\ &\frac{\partial E^*}{\partial t^*} + \nabla^* \cdot \big((E^* + p^*)\mathsf{v}^*\big) = \nabla^* \cdot (\tau^* \cdot \mathsf{v}^* - \mathsf{q}^*) \end{aligned} \tag{1.29}$$

where the non dimensional viscous stress tensor writes

$$\tau^* = \frac{\mu_r}{\rho_r \mathsf{v}_r l} \big(-\frac{2}{3}\mu^* (\nabla^* \cdot \mathsf{v}^*)\mathbb{I} + \mu^* (\nabla^{*t} \mathsf{v}^* + \nabla^* \mathsf{v}^*) \ \big) \tag{1.30}$$

and the non dimensional heat flux writes

$$\mathsf{q}^* = -\frac{\gamma R T_r \mu_r}{\rho_r \mathsf{v}_r^3 l(\gamma-1)P_r}\mu^*\nabla^* T^*. \tag{1.31}$$

As for the Euler equations, we choose as reference speed the free stream speed of sound and as reference temperature and viscosity the free stream temperature and free stream viscosity. This again is a classical choice of nondimensionalization [68].

The non dimensional viscosity therefore writes

$$\mu^* = \frac{\mu}{\mu_\infty} = \mu_0 \frac{T_0+C}{T+C}\frac{T}{T_0}^{\frac{3}{2}} \Big/ \left(\mu_0 \frac{T_0+C}{T_\infty+C}\left(\frac{T_\infty}{T_0}\right)^{\frac{3}{2}}\right) = \frac{1+\frac{C}{T_\infty}}{T^*+\frac{C}{T_\infty}}(T^*)^{\frac{3}{2}} \tag{1.32}$$

and we obtain the non dimensional viscous stress tensor

$$\tau^* = \frac{M_\infty}{Re_\infty}\Big(-\frac{2}{3}\mu^*(\nabla^*\cdot\mathsf{v}^*)\mathbb{I} + \mu^*(\nabla^{*t}\mathsf{v}^* + \nabla^*\mathsf{v}^*)\Big) \tag{1.33}$$

and the non dimensional heat flux

$$\mathsf{q}^* = -\frac{M_\infty}{Re_\infty}\frac{1}{(\gamma-1)P_r}\mu^*\nabla^* T^* \tag{1.34}$$

where

$$Re_\infty = \frac{\rho_\infty \mathsf{v}_\infty l}{\mu_\infty}. \tag{1.35}$$

Finally, with this nondimensionalization process, the free stream values write

$$\rho_\infty^* = 1,\ p_\infty^* = \frac{1}{\gamma},\ E_\infty^* = \frac{1}{\gamma(\gamma-1)},\ \mathsf{v}_\infty^* = \frac{\mathsf{v}_\infty}{c_\infty} = M_\infty \mathbf{n}_\infty,\ T_\infty^* = 1,\ \mu_\infty^* = 1. \tag{1.36}$$

1.6 Discretization with the Residual Distribution method

We present in this section the basic notions of the Residual Distribution schemes. The presentation here is focused on the resolution of a generic hyperbolic conservation equation in the form (1.3). The technical details used to solve the Euler equations with a Residual Distribution scheme are reviewed in chapter 2 and the extension of the Residual Distribution method to the Navier-Stokes equations is presented in chapter 3.

1.6.1 Triangulation and discrete space

We want to solve the equation (1.3) defined on the domain Ω

$$\text{div } \mathbf{f}(\mathbf{u}(x)) = 0, x \in \Omega. \tag{1.37}$$

The first step is to establish a triangulation of $\bar{\Omega}$, which consists in constructing a partition of $\bar{\Omega}$ into a finite number of subsets K that verify the following properties:

(i) $\bar{\Omega} = \cup_{K \in \mathscr{T}_h}$ (to simplify, we suppose that the domain Ω is the interior of the reunion of the elements K, which are either non degenerated triangles or tetrahedrons according to the dimension of Ω),

(ii) for each $K \in \mathscr{T}_h$, K is a closed set with its interior non empty,

(iii) if two elements are distinct then their interiors are disjoint,

(iv) any face of an element $K_1 \in \mathscr{T}_h$ is either the face of another element $K_2 \in \mathscr{T}_h$, or a face on the boundary Ω.

For each element K, we define the set Σ_K of the N distinct points of K defined by

$$\Sigma_K = \left\{ x \in \mathbb{R}^n, \lambda_j(x) \in \{0, \frac{1}{k}, ..., \frac{k-1}{k}, 1\}, 1 \leq j \leq n+1 \right\} \tag{1.38}$$

where $\lambda_j, j = 1, ..., d+1$ are the barycentric coordinates with respect to the vertices of K. We introduce the finite dimension vector space $\mathbb{P}^k(K)$ which contains the polynomial functions of degree k defined on K. Let us now define a basis of $\mathbb{P}^k(K)$. In the case of Lagrange finite elements the basis function $\varphi_i, 1 \leq i \leq N$ is defined by

$$\varphi_i(a_j) = \delta_{ij} \tag{1.39}$$

with $a_j \in \Sigma_K$ and where δ_{ij} is the Kronecker delta function. An other example of basis is given in section 2.

The second step is to define a global approximation space from the local spaces that have been constructed.

We introduce the finite dimension space

$$X_h = \left\{ \mathbf{u}_h \in C^0(\bar{\Omega})^p;\ \forall K \in \mathscr{T}_h,\ \mathbf{u}_{h|K} \in \mathbb{P}^k(K)^p, k \in \mathbb{N}^* \right\} \tag{1.40}$$

that will constitute the global approximation space. We denote by Σ_h the set of cardinal N_h of the nodes of the finite elements:

$$\Sigma_h = \cup_{K \in \mathscr{T}_h} \Sigma_K \tag{1.41}$$

and we remark from the hypothesis (iv) and the definitions 1.38 and 1.39 that the restrictions of the set of nodes and the basis functions of two elements $K_1, K_2 \in \mathscr{T}_h$ coincide at the interface Γ between these two elements. So, a natural basis of X_h is constituted of the basis functions $\varphi_i, 1 \leq i \leq N_h$ of the finite elements $(K, \mathbb{P}^k(K)^p, \Sigma_K), K \in \mathscr{T}_h$.

The space X_h being now equipped with the basis we have defined, we can write the approximate solution $\mathbf{u}_h \in X_h$ of the problem (1.37) as

$$\mathbf{u}_h = \sum_{i=1}^{N_h} \mathbf{u}_i \varphi_i \ . \tag{1.42}$$

1.6.2 Construction of the Residual Distribution scheme

The approximate solution $\mathbf{u}_h$ is not in general a solution of the equation (1.37). Consequently, if in the equation (1.37) we replace the solution $\mathbf{u}$ by the approximate solution $\mathbf{u}_h$, then the right member of the equation (1.37) will be equal to a non zero quantity. So, the idea is to integrate the left-hand side of equation (1.37) and we obtain a quantity called the "total residual" of K

$$\Phi^K(\mathbf{u}_h) = \int_K \text{div } \mathbf{f}(\mathbf{u}_h) dx. \tag{1.43}$$

As we have supposed that the numerical solution $\mathbf{u}_h$ is globally continuous from the definition of the space X_h (1.40), with the Green theorem the volume integral above can be transformed into a surface integral

$$\Phi^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \ dx \tag{1.44}$$

where $\mathbf{n}$ is the exterior normal to ∂K.

In practice, this is the formulation that is used to compute the total residual. In particular, it avoids to compute a numerical approximation of the derivative of $\mathbf{f}(\mathbf{u}_h)$ and guaranties in some sense the numerical conservation of the scheme, as the numerical values of the flux at the quadrature points of a face are common for the two elements sharing that face.

In the same manner, we define for each face (or segment in dimension two) $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$ of the boundary of the domain Ω the total residual of Γ

$$\Phi^\Gamma := \int_\Gamma \big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}_h)\big) \cdot \mathbf{n} \, dx \tag{1.45}$$

where $\mathbf{f}(\mathbf{u}^\Gamma)$ is a numerical flux that depends on the boundary condition $\mathbf{u}^\Gamma$, the outward normal $\mathbf{n}$ and the local state $\mathbf{u}_h$.

Once the total residual Φ^K is computed, the next step is to compute the "nodal residuals" Φ^K_σ (also called "residuals" or "sub-residuals" in the literature) that write in generic form

$$\Phi_\sigma^K = \beta_\sigma^K(\mathbf{u}_h)\Phi^K(\mathbf{u}_h), \sigma \in \Sigma_K. \tag{1.46}$$

Similarly, the boundary nodal residuals Φ_σ^Γ are defined by

$$\Phi_\sigma^\Gamma = \beta_\sigma^\Gamma(\mathbf{u}_h)\Phi^\Gamma(\mathbf{u}_h), \sigma \in \Sigma_\Gamma \tag{1.47}$$

where Σ_Γ is the set of nodes of the face Γ lying on the boundary of the domain. The coefficients β_σ^K et β_σ^Γ are the distribution coefficients and in general they depend on the solution $\mathbf{u}_h$. They are real numbers in the scalar case and are matrices in the case of a system of equations. The nodal residuals must satisfy the conservation constraints

$$\sum_{\sigma\in\Sigma_h} \Phi_\sigma^K = \Phi^K, \forall\ K \in \mathscr{T}_h \tag{1.48}$$

and

$$\sum_{\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = \Phi^\Gamma, \forall\ \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}\ . \tag{1.49}$$

The essential part of the method is the definition of the residuals Φ_σ^K and Φ_σ^Γ, according to the desired properties of the numerical scheme.
As examples of distribution schemes, we give here without going into details, respectively the Lax-Wendroff scheme and the limited stabilized Rusanov scheme in the case of the Euler equations:

$$\Phi_\sigma^{K,LW} = \frac{\Phi^K}{N_{dof}^K} + \int_K \big(A\cdot\nabla\varphi_\sigma\big)\,\Xi\,\big(A\cdot\nabla\mathbf{u}_h\big)dx, \tag{1.50}$$

$$\widehat{\Phi}_\sigma^{K,Rus} = \widehat{\Phi}_\sigma^K + \theta\int_K \big(A\cdot\nabla\varphi_\sigma\big)\,\Xi\,\big(A\cdot\nabla\mathbf{u}_h\big)dx \tag{1.51}$$

where in the Rusanov scheme, the hat symbol denotes the use of a limitation technique and the term θ is a shock capturing term. These schemes are detailed in chapter 2 and their versions for the Navier-Stokes equations are described in chapter 3.

The residuals inside the domain have been defined and we have now to define the boundary residuals.

The boundary residuals Φ_σ^Γ are used to impose weakly the boundary conditions and are defined by the following reasoning.

We start from a weak formulation of the conservation equation (1.3)

$$\int_\Omega \psi\ \mathrm{div}\mathbf{f}(\mathbf{u}_h) = 0 \tag{1.52}$$

where ψ is a regular function. From the Green formula, we have

$$-\int_\Omega \mathbf{f}(\mathbf{u}_h)\cdot\nabla\psi + \int_\Gamma \psi\mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} = 0 \tag{1.53}$$

where here Γ denotes the boundary of Ω. If we take into account the boundary condition $\mathbf{u}_h = \mathbf{u}^\Gamma$ on Γ we obtain

$$-\int_\Omega \mathbf{f}(\mathbf{u}_h)\cdot\nabla\psi + \int_\Gamma \psi\ \mathbf{f}(\mathbf{u}^\Gamma)\cdot\mathbf{n} = 0 \tag{1.54}$$

If we again apply the Green formula, this time to the first term of the left member of the equation above, we have

$$-\int_\Omega \mathbf{f}(\mathbf{u}_h)\cdot\nabla\psi = \int_\Omega \psi\ \mathrm{div}\mathbf{f}(\mathbf{u}_h) - \int_\Gamma \psi\ \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} \tag{1.55}$$

and finally we obtain

$$\int_\Omega \psi\ \mathrm{div}\mathbf{f}(\mathbf{u}_h) + \int_\Gamma \psi(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}_h))\cdot\mathbf{n} = 0. \tag{1.56}$$

If we replace the function ψ by the basis function φ_σ associated to the node σ, we have, by denoting Ω_σ the support of φ_σ

$$\int_{\Omega_\sigma} \varphi_\sigma\ \mathrm{div}\mathbf{f}(\mathbf{u}_h) + \int_{\Gamma\cap\Omega_\sigma} \varphi_\sigma(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}_h))\cdot\mathbf{n} = 0 \tag{1.57}$$

and the second term of the left member can be interpreted as the contribution of the boundary of the domain Ω that must be added to the residuals computed in the elements of the domain [34].

Let us now define the numerical flux $\mathbf{f}(\mathbf{u}^\Gamma)$.

Inflow/outflow boundary conditions
In order to impose the inflow/outflow boundary conditions, we define an inflow/outflow state $\mathbf{u}^\Gamma_{\mathrm{in/out}}$ at the boundary and the advection flux is linearized. This gives this approximation

$$\mathbf{f}^a(\mathbf{u}^\Gamma_{\mathrm{in/out}})\cdot\mathbf{n} \simeq A^+_{\mathbf{n}}(\mathbf{u}_h)\mathbf{u}_h + A^-_{\mathbf{n}}(\mathbf{u}_h)\mathbf{u}^\Gamma_{\mathrm{in/out}}. \tag{1.58}$$

We define now the terms A^+_n et A^-_n.
First, we denote by A the term $A = (A_1, \ldots, A_d)$ where

$$A_i(\mathbf{u}) = \frac{\partial \mathbf{f}^a_i(\mathbf{u})}{\partial \mathbf{u}} \tag{1.59}$$

is the Jacobian matrix of the flux in the direction i. The product $A\cdot\mathbf{n}$ is the matrix defined by

$$A\cdot\mathbf{n} = \sum_{i=1}^{d} A_i\mathbf{n}_i \tag{1.60}$$

and, as we consider the advection flux, the matrix A is diagonalizable in $\mathbb{R}$. We can then define the matrix $(A \cdot \mathbf{n})^+$ by

$$(A \cdot \mathbf{n})^+ = R_{\mathbf{n}} \wedge_{\mathbf{n}}^+ L_{\mathbf{n}} \tag{1.61}$$

where $R_{\mathbf{n}}$ and $L_{\mathbf{n}}$ are respectively the matrices of the right and left eigenvectors of $A \cdot \mathbf{n}$ and $\wedge_{\mathbf{n}}^+$ is the diagonal matrix with the terms $\max(\lambda_{\mathbf{n}}, 0)$ on the diagonal, where the $\lambda_{\mathbf{n}}$ are the eigenvalues of $A \cdot \mathbf{n}$. Similarly, we define the matrix $(A \cdot \mathbf{n})^-$ with $\wedge_{\mathbf{n}}^-$ which is now the diagonal matrix with the terms $\min(\lambda_{\mathbf{n}}, 0)$ on the diagonal.
Finally, by linearizing the interior flux with

$$\mathbf{f}^a(\mathbf{u}_h) \simeq A_{\mathbf{n}}(\mathbf{u}_h)\mathbf{u}_h \tag{1.62}$$

we obtain the following expression of the flux for the imposition of the inflow/outflow boundary condition

$$\left(\mathbf{f}^a(\mathbf{u}^\Gamma_{\text{in/out}}) - \mathbf{f}^a(\mathbf{u}_h)\right) \cdot \mathbf{n} = A^-_{\mathbf{n}}(\mathbf{u}_h)(\mathbf{u}^\Gamma_{\text{in/out}} - \mathbf{u}_h). \tag{1.63}$$

Slip wall boundary conditions
In the case of a non viscous wall, the slip wall boundary condition $\mathbf{u} \cdot \mathbf{n} = 0$ is imposed weakly. With this boundary condition the flux at the boundary writes

$$\mathbf{f}^a(\mathbf{u}^\Gamma_{\text{wall}}) = \begin{pmatrix} 0 \\ p\ \mathbf{n} \\ 0 \end{pmatrix} \tag{1.64}$$

and the expression of the flux for the imposition of the slip wall boundary condition writes

$$\left(\mathbf{f}^a(\mathbf{u}^\Gamma_{\text{wall}}) - \mathbf{f}^a(\mathbf{u}_h)\right) \cdot \mathbf{n} = -\mathsf{v} \cdot \mathbf{n} \begin{pmatrix} \rho \\ \rho\ \mathsf{v} \\ \rho E + P \end{pmatrix}. \tag{1.65}$$

Adiabatic slip wall boundary conditions
Finally, in the case of a viscous wall, we impose weakly the boundary condition "adiabatic no slip", it means that the velocity and the normal heat flux are zero on the boundary

$$\mathsf{v}_{|\text{wall}} = 0,\ \mathsf{q} \cdot \mathbf{n} = 0. \tag{1.66}$$

However, this boundary condition has not been used in the work presented here, indeed in the work presented here the wall boundary conditions are imposed with a penalization term as shown in chapter 3.

Now the internal and boundary residuals are defined and we can write the complete numerical scheme. It is written for each node of Σ_h as a system of non linear equations

$$\sum_{K,\sigma\in\Sigma_K} \Phi_\sigma^K + \sum_{\Gamma,\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = 0,\ \forall\sigma\in\Sigma_h, \tag{1.67}$$

which is in practice solved by an iterative method that writes for example in the case of an explicit scheme

$$\frac{\mathbf{u}_\sigma^{n+1}-\mathbf{u}_\sigma^n}{\Delta t_\sigma^n} + \sum_{K,\sigma\in\Sigma_K} \Phi_\sigma^K(\mathbf{u}_h^n) + \sum_{\Gamma,\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = 0,\ \forall\sigma\in\Sigma_h. \tag{1.68}$$

1.6.3 Properties of the Residual Distribution scheme

Convergence to a weak solution
The Lax-Wendroff theorem for Residual Distribution schemes is presented in section 2. We recall it here for completeness.
It is assumed that if

(i) the family of triangulations $(\mathscr{T}_h)_h$ is regular in the sense defined in (1.6.1),

(ii) the residuals (1.46)-(1.47) depend continuously on $\mathbf{u}_h$,

(iii) the sequence of solutions $(\mathbf{u}_h)_h$ is uniformly bounded for the norm $L^\infty(\Omega)$,

(iv) there exists a function $\mathbf{v}\in L^2(\Omega)^p$ such that there exists a subsequence of $(\mathbf{u}_h)_h$ that converges to $\mathbf{v}$ in $L^2_{\text{loc}}(\Omega)^p$,

then $\mathbf{v}$ is a weak solution of the equation (1.37). The main argument in the proof is the conservation relation (1.48)-(1.49). In the proof of the theorem, the continuity of the interpolant $\mathbf{u}_h$ across adjacent faces is assumed, but this constrain can be weakened, as shown in chapter 2.

Truncation error
As we use unstructured meshes, the truncation error is studied with a weak formulation [5]. So, we suppose that φ is a C^1 test function with a compact support in Ω and we introduce the truncation error for the Residual Distribution schemes

$$\epsilon(\mathbf{w}_h,\varphi) = \sum_{\sigma\in\Sigma_h} \varphi(\sigma)\left(\sum_{K,\sigma\in K} \Phi_\sigma^K + \sum_{\Gamma,\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma\right) \tag{1.69}$$

where $\mathbf{w}_h$ is the interpolant of the exact solution $\mathbf{w}$. If $\mathbf{w}$ is assumed to be regular enough and if the residuals verify the following relations

$$\begin{aligned} &\Phi_\sigma^K(\mathbf{w}_h) = \mathcal{O}(h^{k+d}), \forall\ K\in\mathscr{T}_h, \sigma\in\Sigma_h \\ &\Phi_\sigma^\Gamma(\mathbf{w}_h) = \mathcal{O}(h^{k+d-1}), \forall\ \Gamma\in\mathscr{T}_h\cap\partial\bar{\Omega}, \sigma\in\Sigma_h \end{aligned} \tag{1.70}$$

and if $\mathbf{f}(\mathbf{w}_h)$ is an approximation of order $k+1$ of $\mathbf{f}(\mathbf{w})$, then it is shown [10] that the truncation error verifies the relation

$$|\epsilon(\mathbf{w}^h, \varphi)| \leq C(\varphi, \mathbf{f}, \mathbf{w}) h^{k+1} \tag{1.71}$$

where C is a constant that depends only on φ, $\mathbf{f}$ and $\mathbf{w}$.

With the same assumptions on $\mathbf{w}_h$ we have the following relations

$$\begin{aligned} &\Phi^K(\mathbf{w}_h) = \mathcal{O}(h^{k+d}), \forall\ K \in \mathscr{T}_h \\ &\Phi^\Gamma(\mathbf{w}_h) = \mathcal{O}(h^{k+d-1}), \forall\ \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}. \end{aligned} \tag{1.72}$$

We remark then that if the distribution coefficients β_σ^K (3.30) and β_σ^Γ (3.31) are uniformly bounded with respect to h, then the relations (1.70) are verified. We then have a criterion to construct a high order Residual Distribution scheme. In particular, this analysis allows to deduce what order of quadrature needs to be chosen for the computation of the total residual of an element, a face or a source term in the numerical scheme, more precisely we require that

$$\begin{aligned} &\sum_{e\in\partial K}\sum_{q\in N_{q_e}} \omega_q \mathbf{f}(\mathbf{u}_h(x_q)).\mathbf{n} = \int_{\partial K} \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} + \mathcal{O}(h^{k+d}) \\ &\sum_{q\in N_{q_K}} \omega_q S(\mathbf{u}_h(x_q)) = \int_K S(\mathbf{u}_h) + \mathcal{O}(h^{k+d}) \\ &\sum_{q\in N_{q_\Gamma}} \omega_q \Big(\big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}_h)\big)(x_q)\cdot\mathbf{n} \Big) = \int_\Gamma \big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}_h)\big)\cdot\mathbf{n} + \mathcal{O}(h^{k+d-1}) \end{aligned} \tag{1.73}$$

where N_{q_e}, N_{q_K} and N_{q_Γ} are the set of quadrature points of respectively the edge e, the element K and the boundary Γ and ω_q is the weight associated to the quadrature point q.

Monotonicity preservation

We want now to determine the conditions that need to be satisfied so that the numerical scheme produces non oscillatory solutions. For this purpose we study the case where the equation (1.37) is a scalar equation of unknown u in the homogeneous case $S = 0$ [10]. In this case the residual can be written in the general form

$$\Phi_\sigma^K = \sum_{\substack{\sigma'\in\Sigma_K \\ \sigma'\neq\sigma}} c_{\sigma\sigma'}^K (u_\sigma - u_{\sigma'}) \tag{1.74}$$

and so, if we neglect the boundary conditions, we can write the system of equations (1.67) as

$$\sum_{K,\sigma\in\Sigma_K}\sum_{\substack{\sigma'\in\Sigma_K\\ \sigma'\neq\sigma}} c^K_{\sigma\sigma'}(u_\sigma - u_{\sigma'}) = 0, \forall\sigma\in\Sigma_h. \tag{1.75}$$

The coefficients $c^K_{\sigma\sigma'}$ depend in general on the solution u_h, which makes the system non linear. It is solved by an iterative method, that can be written as

$$u^{n+1}_\sigma = u^n_\sigma - \omega_\sigma\Big(\sum_{K,\sigma\in K}\sum_{\sigma'\in K} c^K_{\sigma\sigma'}(u_\sigma - u_{\sigma'})\Big)^n \tag{1.76}$$

where ω_σ is a relaxation parameter.

If we suppose that the following positivity conditions are satisfied

$$\sum_{K,\sigma,\sigma'\in K} c^K_{\sigma\sigma'} \geq 0 \quad \forall\sigma,\sigma' \quad \text{et} \quad 1-\omega_\sigma\Big(\sum_{K,\sigma\in K}\sum_{\sigma'\in K} c^K_{\sigma\sigma'}\Big) \geq 0 \quad \forall\sigma \tag{1.77}$$

we then show that the solution verifies a discreet maximum principle

$$\min_{K,\sigma\in K}\min_{\sigma'\in K} u^0_{\sigma'} \leq u^n_\sigma \leq \max_{K,\sigma\in K}\max_{\sigma'\in K} u^0_{\sigma'} \tag{1.78}$$

where u^0_σ is the initial value of the solution at the node σ. This implies that the solution is stable for the norm L^∞. In practice, the local positivity conditions are used instead. They are stronger than the conditions defined above and they write

$$c^K_{\sigma\sigma'} \geq 0 \quad \forall\sigma,\sigma'\in K, \forall K \quad \text{et} \quad \omega_\sigma \max_{K,\sigma\in K}\Big(\frac{\mathcal{C}_\sigma}{\mathcal{C}^K_\sigma}\sum_{\sigma'\in K} c_{\sigma\sigma'}\Big) \leq 1 \tag{1.79}$$

where $\mathcal{C}^K_\sigma = \frac{|K|}{d}$ and $\mathcal{C}_\sigma = \sum_{K,\sigma\in\Sigma_K}\mathcal{C}^K_\sigma$. A scheme that verifies these two conditions is called "monotonicity preserving".

The reasoning above cannot be applied in the case of a system of equations as the coefficients $c^K_{\sigma\sigma'}$ are then matrices. However, with the Rusanov scheme (2.29) and the stability criterion invoked in section 2.2.2, the solution obtained is in practice stable.

If now we use an implicit scheme in the case of a scalar equation and if this scheme can be written as

$$u^{n+1}_\sigma = u^n_\sigma - \omega_\sigma\Big(\sum_{K,\sigma\in K}\sum_{\sigma'\in K} c^K_{\sigma\sigma'}(u_\sigma - u_{\sigma'})\Big)^{n+1} \tag{1.80}$$

then it is shown [63] that if the scheme verifies the first condition of (1.77) then the scheme verifies the discrete maximum principle (1.78) unconditionally (with respect to the time step). In the case of a system of equations, the

Rusanov scheme (2.29) is in practice stable.

Monotonicity preservation and high order
The construction of a high order monotonicity preserving Residual Distribution scheme is detailed in chapter 2. The principal idea is to start from a monotone scheme and to apply a limitation procedure to its distribution coefficients, so that the scheme is monotone and high order. The scheme obtained is then improved with a stabilization term.

Chapter 2

Construction of a p-adaptive continuous Residual Distribution scheme for hyperbolic problems

This chapter consists of a journal paper submitted and under revision.

A p-adaptive continuous Residual Distribution scheme is proposed in this chapter. Under certain conditions, primarily the expression of the total residual on a given element K into residuals on the sub-elements of K and the use of a suitable combination of quadrature formulas, it is possible to change locally the degree of the polynomial approximation of the solution. The discrete solution can then be considered non continuous across the interface of elements of different orders, while the numerical scheme still verifies the hypothesis of the discrete Lax-Wendroff theorem which ensures its convergence to a correct weak solution. We detail the theoretical material and the construction of our p-adaptive method in the frame of a continuous Residual Distribution scheme. Different test cases for non-linear equations at different flow velocities demonstrate numerically the validity of the theoretical results.

2.1 Introduction

Because of their potential in delivering higher accuracy with lower cost than low order methods, high-order methods for computational fluid dynamics (CFD) have obtained considerable attention in the past two decades [89]. By high order, we mean third order or higher. Most industrial codes used today for CFD simulations are based upon second-order finite volume methods (FVM), finite difference methods (FDM), or finite element methods (FEM) [89]. As second-order methods are used in most CFD codes, some very complex flows simulations might remain out of their reach. Indeed, in some cases, second-order methods are still too much dissipative and as a consequence they require much finer meshes and become too expensive even on modern supercomputer clusters. In order to deal with a large and diverse range of problems, lots of researches have been conducted with the aim of designing robust and stable high order methods, see [41] and [86]. High-order methods allow the use of coarser meshes [89, 86, 87], high-order boundary representation [73], and improve the accuracy of the solutions [89, 86, 30]. Because of their potential, we believe that the next generation of CFD solvers will have to be based upon high order methods. Besides the use of higher order methods, a very promising approach is the use of *hp*-adaptation to change locally the order of accuracy and the size of the mesh according to the solution [69].

Among high-order methods, the Discontinuous Galerkin (DG) method [59, 38, 79], the Residual Distribution (RD) method [10, 1, 6, 35**?**] and the hp finite element method (*hp*-FEM) [23, 18, 81, 37] have attracted a lot of interest in the recent years.

The DG method has a compact stencil regardless the order of the polynomials representing the solution [54]. This very local formulation leads to a great flexibility, especially for the parallelization of its implementation. However, DG methods suffer from the rapid increase of the number of nodes [29], and then, simulations in three-dimensional space may quickly become too expensive. A possible way to overcome this problem is to use p-adaptation [23] (which is conceptually easy with DG methods), where the local approximation order p (hence the term p-adaptation) is dictated by the flow field. The optimal solution is *hp*-adaptation (mesh and polynomial adaptations) to achieve the best accuracy with the minimum cost. In smooth regions, p-adaptation is preferred, whereas in discontinuous regions, h-adaptation is preferred.

Another possible approach is the class of residual distribution schemes. RD methods have a longer experience in stabilization mechanisms and shock capturing abilities than DG schemes [78]. Moreover, RD methods offer a very compact stencil like DG methods, but with a smaller number of nodes [10]. The drawback is that to achieve this low number of nodes, the continuity of the approximation is required and consequently, the use of p-adaptation with continuous finite elements in the frame of RD schemes is a priori not possible

as this would violate the continuity requirement. The aim of this work is to demonstrate that it is indeed possible to use p-adaptation with continuous finite elements in the frame of RD schemes. Such an approach, while offering the same advantages as classical residual distribution schemes (like among others the low number of nodes, the non-oscillatory behavior and the accuracy on smooth problems [10]), exhibits some interesting advantages thanks to p-adaptation, like an improved convergence and better shock capturing abilities. The practical implementation of p-adaptation for RD schemes results in a residual based solver that can use high order elements in smooth regions, and low order elements in discontinuous regions. In this sense, this approach is following the recommendations for the next generation of CFD solvers [89].

The hp finite element method can be traced back to the work of I. Babuska et al. [23]. The authors presented an evolution of the finite element method that showed super convergent properties thanks to a suitable combination of h-adaptation and p-adaptation. Of particular interest to our work, the hp finite element method applied to the simulation of turbulent and laminar flows presented in [18], is a method that couples dynamic adaptation techniques with a high order streamline/upwind Petrov-Galerkin (SUPG) finite element scheme. This method shares some traits with the work presented here, notably the authors try to achieve similar goals: the construction of a high order adaptive scheme using continuous finite elements for the simulation of fluid dynamics. But, as we will show, the construction presented by the authors is very different to the method we present here for Residual Distribution schemes.

The chapter is organized as follows. In section 2.2, the mathematical problem is defined and we recall briefly the general principles of the Residual Distribution schemes. In section 2.3, we expose how it is theoretically possible to use p-adaptation in the frame of a continuous Residual Distribution scheme and we propose the detailed construction of the p-adaptive RD scheme. We present some numerical results in section 2.4, along with some benefits brought by p-adaptation. In conclusion, we invoke some possible future extensions and developments to the work presented here.

2.2 Mathematical problem and residual distribution schemes

2.2.1 Basic notions of residual distribution schemes

In this section, the basic notions of RD schemes are summarized, more details can be found in [10]. We are interested in the numerical approximation of steady hyperbolic problems of the form

$$\operatorname{div} \mathbf{f}(\mathbf{u}) = 0\,, \tag{2.1}$$

where $\mathbf{u} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^{d+2}$ is the vector of conservative variables, Ω is an open set of $\mathbb{R}^d$, $d = 2, 3$ is the spatial dimension and $\mathbf{f} = (\mathbf{f}_1, ..., \mathbf{f}_d)$ is the vector of flux functions, with $\mathbf{f}_i(\mathbf{u}) : \Omega \to \mathbb{R}^{d+2}$, $i = 1, ..., d$. For convenience, the flux vector $\mathbf{f}$ can be written in column:

$$\mathbf{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_{d+2} \end{pmatrix} \tag{2.2}$$

where f_j=$(\mathbf{f}_{1_j}, \ldots, \mathbf{f}_{d_j})$.

Dirichlet boundary conditions are weakly imposed on the inflow boundary [10]

$$\partial\bar{\Omega}^- = \{x \in \partial\bar{\Omega}, \mathbf{n} \cdot \nabla_{\mathbf{u}}\mathbf{f} < 0\} \tag{2.3}$$

where $\mathbf{n}$ is the outward normal.

We study in the present work the numerical solution of the compressible Euler system of equations with the vector of unknowns

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathsf{v} \\ E \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} \rho \mathsf{v} \\ \rho \mathsf{v} \otimes \mathsf{v} + p \ \mathrm{Id}_{d\times d} \\ (E + p)\mathsf{v} \end{pmatrix}$$

where ρ is the density, v is the fluid velocity, p is the pressure and $E = \rho e^t$ is the total energy per unit volume where e^t is the specific total energy. We consider a calorically perfect gas, and the Euler equations are closed with the relations for energy

$$e^t = e + \frac{\mathsf{v} \cdot \mathsf{v}}{2} \quad \text{and} \quad e = \frac{1}{\gamma - 1} RT, \tag{2.4}$$

with the thermodynamic relation of state for a perfect gas

$$p = \rho RT \tag{2.5}$$

where $\gamma = 1.4$ is the specific heat ratio, T the gas temperature and R the gas constant which is $287 N.m/kg$ for sea-level air.

In this setting, the vector of unknowns $\mathbf{u} \in \mathbb{R}^{2+d}$ is such that $\rho > 0$ and $e > 0$. On solid boundaries, we impose weakly no slip boundary conditions.

In this section, the discussion will stay rather general, and we will not focus particularly on the Euler system, but we are going to work on a generic hyperbolic system.

We want to find an approximate solution to equations (2.1)-(2.3). We suppose in the work presented here that Ω is a polyhedric open set, more precisely we suppose that $\bar{\Omega}$ is constituted of simplices (triangles in dimension two and

tetrahedrons in dimension three). We can then consider a finite decomposition of the domain

$$\bar{\Omega} = \underset{K\in\mathscr{T}_h}{\cup} K \tag{2.6}$$

where K is a simplex and $\mathscr{T}_h$ is a conformal triangulation of $\bar{\Omega}$.
We associate to each simplex $K \in \mathscr{T}_h$ a finite element $(K, \mathbb{P}^k(K)^p, \Sigma_K)$ where Σ_K denotes the set of nodes of K defined by

$$\Sigma_K = \Big\{ x \in \mathbb{R}^n, \lambda_j(x) \in \{0, \frac{1}{k}, ..., \frac{k-1}{k}, 1\}, 1 \leq j \leq n+1 \Big\} \tag{2.7}$$

where $\lambda_j, j = 1, ..., d+1$ are the barycentric coordinates with respect to the vertices of K (the finite elements thus defined are affine equivalent to each other).

For practical applications, it is important to define the basis of the vector space $\mathbb{P}^k(K)$. A natural choice is made by the Lagrange basis. Another choice that will be useful in this work is the Bézier basis. From the definition (3.19), any node $a_\mu \in \Sigma_K$ can be written as

$$a_\mu = \frac{1}{k} \sum_{j=1}^{d+1} \mu_j a_j, \;\; \mu = (\mu_1, ..., \mu_{d+1}) \tag{2.8}$$

where $(a_j)_{j=1,...,d+1}$ are the vertices of K and $\frac{1}{k}(\mu_1, ..., \mu_{d+1})$ are the barycentric coordinates of a_μ with respect to the vertices of K.
The Lagrange basis function φ_μ associated to a_μ can then be written as

$$\varphi_\mu = \big(\prod_{j=1}^{d+1} (\mu_j!) \big)^{-1} \prod_{j=1,\; \mu_j \geq 1}^{d+1} \prod_{i=0}^{\mu_j - 1} \big(k\lambda_j - i \big) \tag{2.9}$$

and the Bézier basis function B_μ associated to a_μ can be written as

$$B_\mu = k! \big(\prod_{j=1}^{d+1} (\mu_j!) \big)^{-1} \prod_{j=1}^{d+1} \lambda_j^{\mu_j} \;\; . \tag{2.10}$$

We remark that the Bézier and Lagrange basis functions sum to unity, and that the Bézier basis functions are positive on K.

We introduce now the global finite element space

$$X_h = \big\{ \mathbf{u}_h \in C^0(\bar{\Omega})^p; \; \forall K \in \mathscr{T}_h, \;\; \mathbf{u}_{h|K} \in \mathbb{P}^k(K)^p, k \in \mathbb{N}^* \big\} \tag{2.11}$$

where we look for the discrete solution to equations (2.1)-(2.3). We denote by Σ_h the set of N_h nodes of the finite elements

$$\Sigma_h = \underset{K\in\mathscr{T}_h}{\cup} \Sigma_K \tag{2.12}$$

and from the conformity assumption of the triangulation, a basis of X_h is constituted of the basis functions (Lagrange or Bézier) $\psi_i, 1 \leq i \leq N_h$ of the finite elements $(K, \mathbb{P}^k(K)^p, \Sigma_K), K \in \mathscr{T}_h$.

We can now write the approximate solution $\mathbf{u}_h \in X_h$ as

$$\mathbf{u}_h = \sum_{i \in \Sigma_h} \mathbf{u}_i \psi_i \ . \tag{2.13}$$

In the Lagrange case, we have $\mathbf{u}_j = \mathbf{u}_h(a_j)$, where $a_j \in \Sigma_h$ is the node associated with the global shape function ψ_j as described above.

In this chapter, all the numerical applications will be made with simplices, but the work presented here can be extended to other kinds of geometric elements like hexahedrons.

We can now write the discrete equations of the residual distribution scheme. We first introduce the total residual of the element K

$$\Phi^K(\mathbf{u}_h) = \int_K \nabla \cdot \mathbf{f}(\mathbf{u}_h) dx, \tag{2.14}$$

which is transformed into the following surface integral with the Green theorem as $\mathbf{u}_h \in X_h$ is continuous

$$\Phi^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \ dx \tag{2.15}$$

where $\mathbf{n}$ is the exterior normal to ∂K. This is the form that is used in practice. We define as well the boundary total residual for a boundary element (edge in dimension two and face in dimension three) $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$

$$\Phi^\Gamma = \int_\Gamma \big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}^h)\big) \cdot \mathbf{n} \, dx \ . \tag{2.16}$$

Once the total residual Φ^K is computed, the next step is to compute the "nodal residuals" Φ^K_σ (also called "residuals" or "sub-residuals") that write in generic form

$$\Phi^K_\sigma = \beta^K_\sigma(\mathbf{u}_h)\Phi^K(\mathbf{u}_h), \sigma \in \Sigma_K. \tag{2.17}$$

Similarly, the boundary nodal residuals Φ^K_σ are defined by

$$\Phi^\Gamma_\sigma = \beta^\Gamma_\sigma(\mathbf{u}_h)\Phi^\Gamma(\mathbf{u}_h), \sigma \in \Sigma_\Gamma \tag{2.18}$$

where Σ_Γ is the set of nodes of the face Γ lying on the boundary of the domain. The coefficients β^K_σ et β^Γ_σ are the distribution coefficients and in general they depend on the solution $\mathbf{u}_h$. They are real numbers in the scalar case and are

matrices in the case of a system of equations. The nodal residuals must satisfy the conservation constraints

$$\sum_{\sigma\in\Sigma_h} \Phi_\sigma^K = \Phi^K, \forall\ K \in \mathscr{T}_h \tag{2.19}$$

and

$$\sum_{\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = \Phi^\Gamma, \forall\ \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}\ . \tag{2.20}$$

The total residual is written with a slight abuse of notation as the integral of $\mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n}$ *which is a vector valued function of size* $d+2$*. With the notations of section 2.2.1, the total residual is then a vector of size* $d+2$ *where the component* $j, 1 \leq j \leq d+2$*, is the integral of the scalar product* $f_j \cdot \mathbf{n}$*. Like the total residual, the nodal residuals are vectors of dimension* $d+2$*. The boundary total and nodal residuals are of the same dimension.*

Finally, the residual distribution scheme writes

$$\forall\,\sigma \in \Sigma_h,\ \sum_{K,\,\sigma\in\Sigma_K} \Phi_\sigma^K + \sum_{\Gamma,\,\sigma\in\Sigma_\Gamma} \Phi_\sigma^\Gamma = 0. \tag{2.21}$$

Before going further, let us give some examples of residual distribution schemes.

2.2.2 Some particular residual distribution schemes

The Lax-Wendroff scheme

The Lax-Wendroff scheme is a central linear scheme, which name comes from the fact that in its scalar version, in the case of $\mathbb{P}^1$ interpolation and for a constant advection speed, the scheme coincides with the scheme named Lax-Wendroff presented in [74]. It is defined by

$$\Phi_\sigma^{K,LW} = \frac{\Phi^K}{N_{dof}^K} + \int_{\Omega_K} \big(A\cdot\nabla\varphi_\sigma\big)\,\Xi\,\big(A\cdot\nabla\mathbf{u}_h\big)dx\ . \tag{2.22}$$

The scalar N_{dof}^K is the cardinal of Σ_K, the term A represents $A = (A_1, \ldots, A_d)$ where

$$A_i(\mathbf{u}) = \frac{\partial \mathbf{f}_i(\mathbf{u})}{\partial \mathbf{u}} \tag{2.23}$$

is the Jacobian matrix of the i-th component of the flux. The product $A\cdot\nabla\varphi_\sigma$ is the matrix defined by

$$A\cdot\nabla\varphi_\sigma = \sum_{i=1}^{d} A_i \frac{\partial\varphi_\sigma}{\partial x_i} \tag{2.24}$$

where φ_σ is the basis function (Lagrange or Bézier, see section 2.2.1) associated to the node σ. The matrix Ξ is a scaling matrix defined by

$$\Xi = \frac{1}{d}|K|\left(\sum_{\sigma\in\Sigma_K} R_{\mathbf{n}_\sigma}(\bar{\mathbf{u}}) \wedge^+_{\mathbf{n}_\sigma}(\bar{\mathbf{u}}) L_{\mathbf{n}_\sigma}(\bar{\mathbf{u}})\right)^{-1}. \tag{2.25}$$

In the definition above $|K|$ is the volume of the element K, $\bar{\mathbf{u}}$ is the average value of the vector of conservative variables on the element K:

$$\bar{\mathbf{u}} = \frac{1}{N^K_{dof}} \sum_{\sigma\in\Sigma_K} \mathbf{u}_\sigma \ , \tag{2.26}$$

$R_{\mathbf{n}_\sigma}$ and $L_{\mathbf{n}_\sigma}$ are respectively the matrices of the right and left eigenvectors of the matrix $A\cdot\mathbf{n}_\sigma$ defined as for (2.24), $\wedge^+_{\mathbf{n}_\sigma}$ is the diagonal matrix with $\max(\lambda_{\mathbf{n}_\sigma}, 0)$ on the diagonal, where the $\lambda_{\mathbf{n}_\sigma}$ are the eigenvalues of the matrix $A\cdot\mathbf{n}_\sigma$. The vector $\mathbf{n}_\sigma$ is defined by

$$\mathbf{n}_\sigma = \frac{1}{d}\int_{\partial K} \nabla\varphi_\sigma \, dx. \tag{2.27}$$

The nodal residuals defined in (2.22) satisfy the conservation relation (2.19) as we have

$$\sum_{\sigma\in\Sigma_K} \nabla\varphi_\sigma = 0 \tag{2.28}$$

for the Lagrange (2.9) and Bézier (2.10) basis functions.

The Rusanov scheme

The Rusanov scheme is a generalization of the one dimensional Rusanov scheme. This scheme is obtained from a central distribution of the total residual with a dissipation term added. In the case of a system of equations it is defined by

$$\Phi^{K,Rsv}_\sigma = \frac{\Phi^K}{N^K_{dof}} + \frac{1}{N^K_{dof}}\alpha^K \sum_{\sigma_j\in\Sigma_K} (\mathbf{u}_\sigma - \mathbf{u}_{\sigma_j}). \tag{2.29}$$

The parameter α^K is chosen as to be larger than the maximum of the spectral radius of the matrix $A\cdot\nabla\varphi_\sigma$. In the scalar case, this choice guaranties that the scheme satisfies a local maximum principle. In the system case, it is proved to be non oscillatory and very dissipative [3].

2.2.3 Construction of a high order monotonicity preserving Residual Distribution scheme

If the flow is smooth, we can use the Lax-Wendroff scheme presented above. However, when the solution presents discontinuities, special care has to be taken to handle them. We recall now the method we follow.

Limitation

From the Godunov theorem [36], a linear scheme cannot be more than first order and monotone. In [3], the idea is to start from a first order monotone scheme and to apply a limitation procedure to it so as to obtain a scheme that is high order and monotone. We choose to apply the limitation procedure to the Rusanov scheme (2.29), as the resulting scheme is well suited to handle the shocks occurring in high speed flows [3].

The limitation procedure is achieved through the following sequence of operations. First, we compute the matrices L_n and R_n which are respectively the matrices of the left and right eigenvectors of the matrix

$$A(\bar{\mathbf{u}}) \cdot \frac{\bar{\mathbf{v}}}{||\bar{\mathbf{v}}||} \tag{2.30}$$

where $A(\bar{\mathbf{u}})$ represents the Jacobian matrix (3.36) evaluated at the average state (2.26) and $\bar{\mathbf{v}}$ is the average speed vector computed the same way as $\bar{\mathbf{u}}$. The nodal residuals (2.29) are projected on the vector space generated by the left eigenvectors L_n, so the intermediate nodal residuals write

$$\Phi_\sigma^{K,*} = L_n \Phi_\sigma^{K,Rsv} \tag{2.31}$$

and we have the total residual

$$\Phi^{K,*} = \sum_{\sigma \in \Sigma_K} \Phi_\sigma^{K,*} \ . \tag{2.32}$$

Then, the high order distribution coefficients are computed from the original first order distribution coefficients by the non linear mapping:

$$\hat{\beta}_\sigma^{K,*} = \frac{\left(\dfrac{\Phi_\sigma^{K,*}}{\Phi^{K,*}}\right)^+}{\displaystyle\sum_{\sigma_j \in K} \left(\frac{\Phi_{\sigma_j}^{K,*}}{\Phi^{K,*}}\right)^+} \ . \tag{2.33}$$

We have built the intermediate high order nodal residuals in the characteristic space:

$$\hat{\Phi}_\sigma^{K,*} = \hat{\beta}_\sigma^{K,*} \Phi^{K,*} \tag{2.34}$$

and finally we project them from the characteristic space to the physical space:

$$\hat{\Phi}_\sigma^{K} = R_n \hat{\Phi}_\sigma^{K,*} \tag{2.35}$$

where R_n are the right eigenvectors defined above.

Stabilization

The solution obtained with a limited Rusanov scheme can exhibit spurious modes and can show poor iterative convergence [10]. These problems are solved via the addition of the filtering term presented above for the scheme (3.39) and so the high order filtered Rusanov scheme for the Euler equations reads

$$\widehat{\Phi}_\sigma^{K,Rus}(\mathbf{u}_h) = \widehat{\Phi}_\sigma^K(\mathbf{u}_h) + \theta \int_{\Omega_K} \big(A \cdot \nabla \varphi_\sigma\big)\, \Xi\, \big(A \cdot \nabla \mathbf{u}_h\big) dx \tag{2.36}$$

where θ is a shock capturing term. In most applications, we take $\theta = 1$, and in some cases a more elaborated version must be chosen. In this work, we take $\theta = 1$.

2.2.4 Boundary conditions

For the nodes lying on the boundary of the domain, we use the following formula for the nodal residuals:

$$\forall\, \Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega},\ \forall \sigma \in \Sigma_\Gamma,\ \Phi_\sigma^\Gamma(\mathbf{u}_h) = \int_\Gamma \varphi_\sigma \big(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_h)\cdot \mathbf{n}\big)\, dx \tag{2.37}$$

where $\mathbf{u}_-$ is the state that is imposed by the Dirichlet conditions and $\mathcal{F}$ is a numerical flux that depends on $\mathbf{u}_-$, the outward normal $\mathbf{n}$ and the local state $\mathbf{u}_h$. In the case of a no-slip condition, the boundary condition $\mathbf{u}_-$ verifies the condition, and the numerical flux is only a pressure flux. More details can be found in [10].

2.2.5 A Lax-Wendroff like theorem and its consequences

The following theorem has been proved in [12]:

Théorème 2.1. *Assume the family of meshes $\mathscr{T} = (\mathscr{T}_h)_h$ is regular. For K an element or a boundary element of $\mathscr{T}_h$, we assume that the residuals $\{\Phi_\sigma^K\}_{\sigma \in K}$ satisfy:*

- *For any $M \in \mathbb{R}^+$, there exists a constant C which depends only on the family of meshes $\mathcal{T}_h$ and M such that for any $\mathbf{u}_h \in X_h$ with $||\mathbf{u}_h||_\infty \leq M$, then*

$$\|\Phi_\sigma^K(\mathbf{u}_{h|K})\| \leq C \sum_{\sigma,\sigma' \in \Sigma_K} |\mathbf{u}_\sigma - \mathbf{u}_{\sigma'}| \tag{2.38}$$

- *They satisfy the conservation property (2.19)-(2.20).*

Then if there exists a constant C_{max} such that the solutions of the scheme (3.37) satisfy $||\mathbf{u}_h||_\infty \leq C_{max}$ and a function $\mathbf{V} \in L^2(\Omega)^p$ such that $(\mathbf{u}_h)_h$ (or at least a sub-sequence) verifies $lim_h||\mathbf{u}_h - \mathbf{v}||_{L^2_{loc}(\Omega)^p} = 0$, then $\mathbf{v}$ is a weak solution of (2.1).

One of the interests of this result, besides indicating automatic consistency constraints for a scheme (2.21), is that the important constraint is to have the conservation relations (2.15)-(2.19) *at the element level.*

Let us take a closer look at the relation (2.15)-(2.19). We need to see how they are actually implemented. Indeed, in the numerical implementation, we do not ask for (2.15)-(2.19), but for a discrete version of these relations, namely:

For any element $K \in \mathscr{T}_h$,

$$\sum_{\sigma \in \Sigma_K} \Phi_\sigma^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \tag{2.39a}$$

and any boundary element $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$,

$$\sum_{\sigma \in \Sigma_\Gamma} \Phi_\sigma^\Gamma = \int_\Gamma \left(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}\left(\mathbf{u}_h\right) \cdot \mathbf{n}\right) dx \tag{2.39b}$$

where here $\int$ signifies that the integral is computed with a quadrature formula, which reads

1. On elements:
$$\int_{\partial K} \mathbf{f}(\mathbf{u}^h) \cdot \mathbf{n} = \sum_{e \text{ edge/face } \subset \partial K} \left(|e| \sum_q \omega_q \mathbf{f}(\mathbf{u}_q) \cdot \mathbf{n} \right)$$
2. On boundary elements:
$$\int_\Gamma \left(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n}\right) dx = |\Gamma| \sum_q \omega_q \left(\mathcal{F}(\mathbf{u}_q, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_q) \cdot \mathbf{n}\right)$$

If we have a close look at the proof of theorem 2.1, we see that besides the boundedness of the sequence of solutions in suitable norms that enables to use compactness argument, what really matters at the algebraic level is that we have the following property on any edge:
If $\Gamma = \Gamma'$ is the same face shared by respectively the two adjacent elements K and K', then we have $\mathbf{n}_{|\Gamma} = -\mathbf{n}_{\Gamma'}$ and consequently

$$\int_\Gamma \varphi_\sigma \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \quad + \int_{\Gamma'} \varphi_\sigma \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} = 0, \tag{2.40}$$

where $\varphi_\sigma \in \mathbb{P}^k(K)$ is the basis function associated to the node σ.
This is clearly true because $\mathbf{u}_h$ is continuous. Now, in the numerical implementation, the easiest way to do so is that the quadrature points on Γ seen from K are the same as the ones on Γ seen from K'. These two remarks are at the core of the present development, as we see now.

2.3 Residual Distribution schemes and *p*-adaptation

As we have seen in the previous section, what matters for conservation is that, for any element K, the sum of the residuals is equal to the total residual, or more precisely

$$\sum_{\sigma \in K} \Phi_\sigma^K = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n}.$$

The total residual $\Phi^K = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n}$ is obtained thanks to quadrature formulas. We show in this section that in many cases this quantity can be rewritten as a weighted sum of total residuals on sub-elements. To make this more precise, we first look at the quadratic case on a triangle, where the Simpson formula is used on each edge. The main idea of this chapter comes from this observation.
We then present the quadratic case on a tetrahedron. From these particular cases, we present two formulas that generalize to arbitrary polynomial orders the formulas given for quadratic polynomials. These general formulas make it possible to construct a p-adaptive Residual Distribution scheme of arbitrary orders, with the definition of a modified nodal residual that we present at the end of the section. We choose to present the particular cases of quadratic approximation along with their generalization, instead of giving only the general formulas, as we think that it is necessary for the comprehension of the reader.

As the formulas presented in this section allow to combine in the same mesh different polynomial orders for the approximation of the solution, we use the term p-adaptation. Our method of p-adaptation is proposed within the frame of a continuous Residual Distribution scheme and is not as general as the method described for example in [18] which is proposed in the frame of the Finite Element method (see Appendix 2.B).

2.3.1 Quadratic interpolation on triangular elements

Let us now consider the case of a triangular element and a quadratic approximation.

We consider a triangle and for simplicity its vertices are denoted by 1,2,3 and the mid-points of the edges are denoted by 4,5,6 (see figure 2.1).

We subdivide the triangle $K = (1, 2, 3)$ into four sub-triangles: $K_1 = (1, 4, 6)$, $K_2 = (4, 2, 5)$, $K_3 = (5, 3, 6)$ and $K_4 = (4, 5, 6)$, as shown in figure 2.1. If we denote by $\lambda_i, i = 1, 2, 3$ the barycentric coordinates corresponding respectively to the vertices $i = 1, 2, 3$ then the linear interpolant of the flux $\mathbf{f}$ writes

$$\mathbf{f}^{(1)} = \sum_{i=1}^{3} \mathbf{f}(\mathbf{u}_i)\lambda_i$$

and the quadratic interpolant of the same flux writes

$$\mathbf{f}^{(2)} = \sum_{i=1}^{6} \mathbf{f}(\mathbf{u}_i)\varphi_i$$

with, from (2.9):

$$\begin{array}{lll} \varphi_i = (2\lambda_i - 1)\lambda_i, & \text{for } i = 1,2,3 & \\ \varphi_4 = 4\lambda_2\lambda_1, & \varphi_5 = 4\lambda_3\lambda_2, & \varphi_6 = 4\lambda_1\lambda_3\,. \end{array}$$

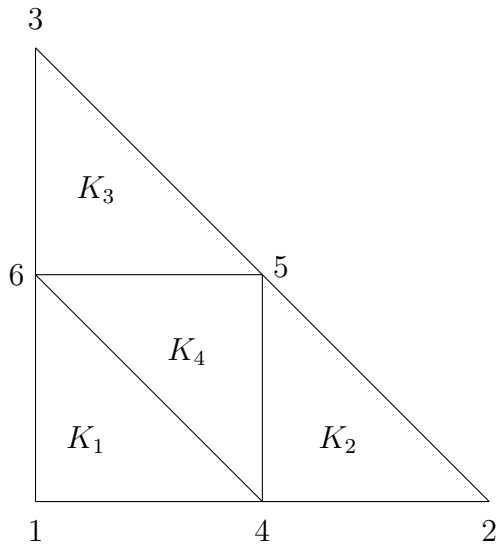


Figure 2.1: Subdivided triangle K.

Let us evaluate the total residual for the quadratic interpolant, we denote

by $\mathbf{n}_i$ the integral $\int_K \nabla\lambda_i dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned}
\int_K \text{div } \mathbf{f}^{(2)} dx &= \sum_{i=1}^{6} \int_K \mathbf{f}_i \cdot \nabla\varphi_i dx \\
&= \sum_{i=1}^{6} \mathbf{f}_i \cdot \int_K \nabla\varphi_i dx \\
&= \sum_{i=1}^{3} \mathbf{f}_i \frac{\mathbf{n}_i}{3} + \frac{4}{3}\big((\mathbf{f}_4(\mathbf{n}_1+\mathbf{n}_2) + \mathbf{f}_5(\mathbf{n}_2+\mathbf{n}_3) + \mathbf{f}_6(\mathbf{n}_1+\mathbf{n}_3) \big) \\
&= \frac{2}{3}\big(\mathbf{f}_1\frac{\mathbf{n}_1}{2} + \mathbf{f}_4\frac{\mathbf{n}_2}{2} + \mathbf{f}_6\frac{\mathbf{n}_3}{2}\big) \\
&+ \frac{2}{3}\big(\mathbf{f}_4\frac{\mathbf{n}_1}{2} + \mathbf{f}_2\frac{\mathbf{n}_2}{2} + \mathbf{f}_5\frac{\mathbf{n}_3}{2}\big) \\
&+ \frac{2}{3}\big(\mathbf{f}_6\frac{\mathbf{n}_1}{2} + \mathbf{f}_5\frac{\mathbf{n}_2}{2} + \mathbf{f}_3\frac{\mathbf{n}_3}{2}\big) \\
&- 2\big(\mathbf{f}_5\frac{\mathbf{n}_1}{2} + \mathbf{f}_6\frac{\mathbf{n}_2}{2} + \mathbf{f}_4\frac{\mathbf{n}_3}{2}\big) \\
&= \frac{2}{3}\Big(\int_{K_1} \text{div } \mathbf{f}^{(1)} dx + \int_{K_2} \text{div } \mathbf{f}^{(1)} dx \\
&\qquad + \int_{K_3} \text{div } \mathbf{f}^{(1)} dx \Big) + 2\int_{K_4} \text{div } \mathbf{f}^{(1)} dx
\end{aligned} \tag{2.41}$$

where $\mathbf{f}^{(1)}$ denotes the $\mathbb{P}^1$ interpolant of the flux in each of the sub-triangles of figure 2.1. The change in signs comes from the fact that the inward normals of the sub-triangle K_4, appearing in the expression of $\int_{K_4} \text{div } \mathbf{f}^{(1)} dx$, are the opposite of the vectors $\mathbf{n}_i/2$.

The relation (2.41) demonstrates that a very simple relation, with positive weights, exists between the $\mathbb{P}^1$ residuals in the sub-triangles and the quadratic residual in K.

2.3.2 Quadratic approximation on tetrahedral elements

The formula (2.41) has no equivalent in dimension three if we use Lagrange finite elements, but this problem can be solved by a change of basis as explained in the following.

For better clarity, we detail the reference quadratic tetrahedron K, described in figure 2.2. The ten nodes of K are numbered 1 to 10 and their coordinates are given by:

$$1 = (0,0,0); \quad 2 = (1,0,0); \quad 3 = (0,0,1); \quad 4 = (0,1,0);$$

$$5 = (1/2, 0, 0); \quad 6 = (1/2, 0, 1/2); \quad 7 = (0, 0, 1/2);$$
$$8 = (1/2, 1/2, 0); \quad 9 = (0, 1/2, 1/2); \quad 10 = (0, 1/2, 0).$$

We subdivide the tetrahedron into height sub-tetrahedrons as shown in figure 2.2. The central octahedron $(5, 6, 7, 8, 9, 10)$ can be split with the diagonal $(7, 8)$, $(6, 10)$ or $(5, 9)$, as shown in figure 2.3, and the sub-tetrahedrons obtained are then:

- Exterior tetrahedrons:
  $K_1 = (5, 7, 10, 1)$; $K_2 = (2, 6, 5, 8)$; $K_3 = (3, 9, 6, 7)$; $K_4 = (8, 9, 10, 4)$.

- With diagonal $(7, 8)$:
  $K_5 = (6, 9, 8, 7)$; $K_6 = (8, 9, 10, 7)$; $K_7 = (8, 10, 5, 7)$; $K_8 = (8, 6, 5, 7)$.

- With diagonal $(6, 10)$:
  $K_5 = (6, 9, 8, 7)$; $K_6 = (8, 9, 10, 7)$; $K_7 = (8, 10, 5, 7)$; $K_8 = (8, 6, 5, 7)$.

- With diagonal $(5, 9)$:
  $K_5 = (5, 9, 6, 7)$; $K_6 = (5, 9, 10, 7)$; $K_7 = (5, 9, 6, 8)$; $K_8 = (5, 9, 10, 8)$.

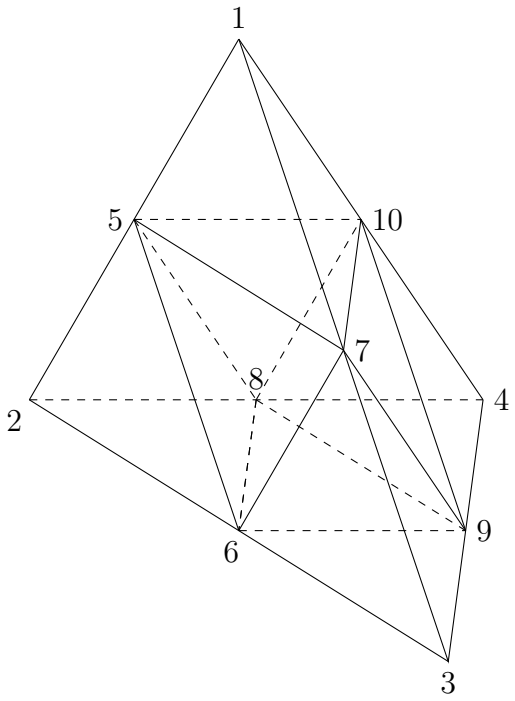


Figure 2.2: Subdivided tetrahedron K.

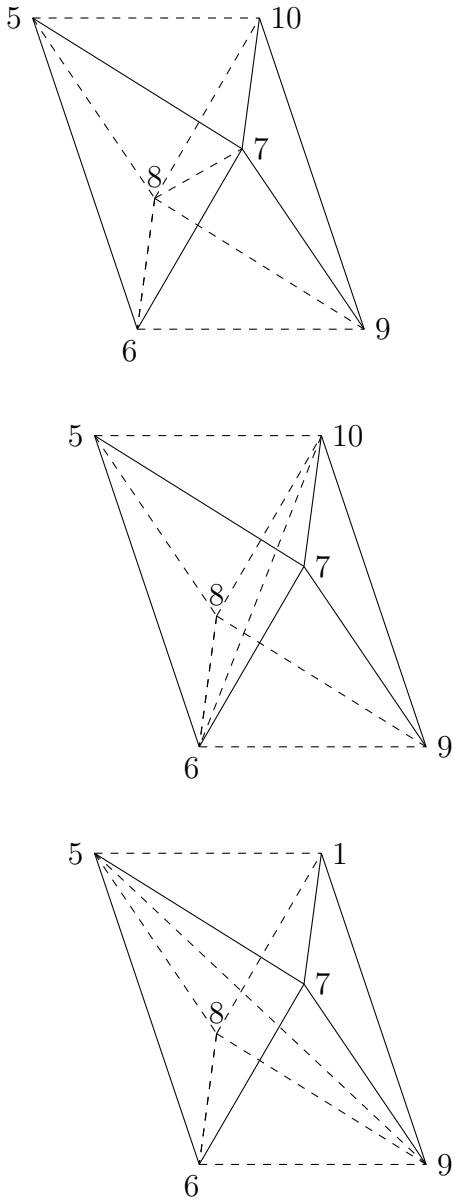


Figure 2.3: Subdivided octahedron with resp. diagonal $(8,7)$,$(10,6)$ and $(5,9)$.

Now if we try to compute the equivalent of formula (2.41) for the three dimensional case using a Lagrange interpolation, we find that the coefficients corresponding to the sub-tetrahedrons K_1, K_2, K_3, K_4 are equal to 0. This is due to a property of the $\mathbb{P}^2$ Lagrange basis functions in dimension three:

$$\int_K \frac{\partial \varphi_j}{\partial x_d} = 0, \quad j = 1, ..., 4, \quad d = 1, 2, 3.$$

This implies that the nodal residuals corresponding to the vertices of the tetrahedron K will not contribute to equation (2.21), which may give a problematic residual scheme.

To avoid this problem, while still using quadratic elements, we use the Bézier basis functions (2.10). More precisely, instead of using a Lagrange interpolation of the flux, we approximate the flux with a Bézier expansion. From the linear algebra point of view, this is a change of basis, but with this basis we obtain positive weights.

If we denote by Φ a quadratic polynomial, with the notations of definition (2.10), the expansion of Φ in the Bézier basis writes

$$\Phi = \sum_{\sigma=1}^{10} \Phi_\sigma B_\sigma$$

and the difference with Lagrange interpolation is that $\Phi_\sigma = \Phi(\sigma)$ only for $\sigma = 1, 2, 3, 4$. For the other nodes, we have

$$\Phi(\sigma) = \frac{\Phi_{j_1} + \Phi_{j_2}}{4} + \frac{\Phi_\sigma}{2} \tag{2.42}$$

where j_1 and j_2 are the two vertices of the tetrahedron on the edge where σ lies. It is clear that $\Phi_\sigma = \Phi(\sigma) + O(h^2)$. We also notice that

$$\nabla B_\sigma = \sum_{j=1}^{4} P_j(\lambda_1, \lambda_2, \lambda_3, \lambda_4) \nabla \lambda_j$$

where $P_j(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ is a polynomial in λ_1, λ_2, λ_3, and λ_4 with *positive* coefficients.

Now we need to find an equivalent formula of (2.41) for the three dimensional case with the Bézier basis functions. In the same spirit as for (2.41), we expand the flux in term of Bézier polynomials:

$$\mathbf{f}^B := \sum_{\sigma=1}^{10} \mathbf{f}_\sigma B_\sigma$$

where for $\sigma = 1, ..., 4$, $\mathbf{f}_\sigma$ is the value of the flux at the vertices of the tetrahedron and when $\sigma > 4$, $\mathbf{f}_\sigma$ is defined according to (2.42).

With the notation $N_i = \nabla \lambda_i$, the gradients of the Bézier basis functions write:

$$\begin{aligned} \nabla B_1 = 2\lambda_1 N_1; \quad \nabla B_2 = 2\lambda_2 N_2; \quad \nabla B_3 = 2\lambda_3 N_3; \quad \nabla B_4 = 2\lambda_4 N_4; \\ \nabla B_5 = 2\big(\lambda_1 N_2 + \lambda_2 N_1\big); \quad \nabla B_6 = 2\big(\lambda_2 N_3 + \lambda_3 N_2\big); \\ \nabla B_7 = 2\big(\lambda_1 N_3 + \lambda_3 N_1\big); \quad \nabla B_8 = 2\big(\lambda_2 N_4 + \lambda_4 N_2\big); \\ \nabla B_9 = 2\big(\lambda_3 N_4 + \lambda_4 N_3\big); \quad \nabla B_{10} = 2\big(\lambda_1 N_4 + \lambda_4 N_1\big) \ . \end{aligned} \tag{2.43}$$

With $\mathbf{n}_j = \int_K N_j$, we find the following formula for the three dimensional case:

$$
\begin{aligned}
2\int_K \text{div } \mathbf{f}^B dx &= 2\sum_{\sigma=1}^{10} \Big(\int_K \mathbf{f}_\sigma \text{div} B_\sigma dx \Big) \\
&= (\mathbf{f}_1\mathbf{n}_1 + \mathbf{f}_2\mathbf{n}_2 + \mathbf{f}_3\mathbf{n}_3 + \mathbf{f}_4\mathbf{n}_4) \\
&+ (\mathbf{f}_5\mathbf{n}_2 + \mathbf{f}_5\mathbf{n}_1) + (\mathbf{f}_6\mathbf{n}_3 + \mathbf{f}_6\mathbf{n}_2) \\
&+ (\mathbf{f}_7\mathbf{n}_1 + \mathbf{f}_7\mathbf{n}_3) + (\mathbf{f}_8\mathbf{n}_4 + \mathbf{f}_8\mathbf{n}_2) \\
&+ (\mathbf{f}_9\mathbf{n}_3 + \mathbf{f}_9\mathbf{n}_4) + (\mathbf{f}_{10}\mathbf{n}_4 + \mathbf{f}_{10}\mathbf{n}_1) \\
&= (\mathbf{f}_1\mathbf{n}_1 + \mathbf{f}_5\mathbf{n}_2 + \mathbf{f}_{10}\mathbf{n}_4 + \mathbf{f}_7\mathbf{n}_3) \qquad (I) \\
&+ (\mathbf{f}_2\mathbf{n}_2 + \mathbf{f}_6\mathbf{n}_3 + \mathbf{f}_5\mathbf{n}_1 + \mathbf{f}_8\mathbf{n}_4) \qquad (II) \\
&+ (\mathbf{f}_3\mathbf{n}_3 + \mathbf{f}_9\mathbf{n}_4 + \mathbf{f}_6\mathbf{n}_2 + \mathbf{f}_7\mathbf{n}_1) \qquad (III) \\
&+ (\mathbf{f}_8\mathbf{n}_2 + \mathbf{f}_9\mathbf{n}_3 + \mathbf{f}_{10}\mathbf{n}_1 + \mathbf{f}_4\mathbf{n}_4) \qquad (IV)
\end{aligned}
\tag{2.44}
$$

The quantities (I), (II), (III), (IV) are interpreted as the integrals of the divergence of the following one degree polynomial functions:

$$
\begin{aligned}
(I) &= \textstyle\int_{K_1} \text{div } \tilde{\mathbf{f}}^{(1)} dx \\
(II) &= \textstyle\int_{K_2} \text{div } \tilde{\mathbf{f}}^{(1)} dx \\
(III) &= \textstyle\int_{K_3} \text{div } \tilde{\mathbf{f}}^{(1)} dx \\
(IV) &= \textstyle\int_{K_4} \text{div } \tilde{\mathbf{f}}^{(1)} dx
\end{aligned}
$$

where

- On K_1, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_1\lambda_1 + \mathbf{f}_5\lambda_2 + \mathbf{f}_{10}\lambda_4 + \mathbf{f}_7\lambda_3$,
- on K_2, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_2\lambda_2 + \mathbf{f}_6\lambda_3 + \mathbf{f}_5\lambda_1 + \mathbf{f}_8\lambda_4$,
- on K_3, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_3\lambda_3 + \mathbf{f}_9\lambda_4 + \mathbf{f}_6\lambda_2 + \mathbf{f}_7\lambda_1$,
- on K_4, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_8\lambda_2 + \mathbf{f}_9\lambda_3 + \mathbf{f}_{10}\lambda_1 + \mathbf{f}_4\lambda_4$.

Strictly speaking, $\mathbf{f}_\sigma$ is *not* the value of the flux at the vertices of K_j, but this is not a problem, following [14].

The equality (2.44) shows like equality (2.41) a relation between the quadratic residual in K and the affine residuals in the sub-tetrahedrons, with positive weights independent of the splitting of the central octahedron.

2.3.3 General subdivision formulas for all orders of approximation

The formulas presented above for the specific case of quadratic approximation can be generalized to any degree of approximation as we show now. For dimension two and dimension three, we use a Bézier basis to obtain consistent

formulas for both cases. The derivation of these formulas is quiet technical, so in order to eliminate any ambiguity, we detail exactly the construction of the formulas with the explaining figures, for dimension two and for dimension three, at the risk of being repetitive. A lot of indices are required for the presentation of the formulas and we notify the reader that the notations used here are local to this subsection. With a slight abuse of notation, in this section the term $\mathbb{P}^k$ triangle (respectively $\mathbb{P}^k$ tetrahedron) represents the finite element $(K, \mathbb{P}^k, \Sigma_K)$ where K is a triangle (respectively a tetrahedron). We denote $\mathbf{f}^B$ the k^{th}-order Bézier expansion of the flux $\mathbf{f}$:

$$\mathbf{f}^B = \sum_{\sigma \in \Sigma_K} \mathbf{f}_\sigma B_\sigma. \tag{2.45}$$

The two dimensional case

For the $\mathbb{P}^k$ triangle, the sub-triangles are numbered from the top to the bottom of the triangle as shown in figure 2.5. This numbering is consistent with the iterative construction of the $\mathbb{P}^k$ triangle from the $\mathbb{P}^{k-1}$ triangle: a set of $\mathbb{P}^1$ sub-triangles is added at the bottom of the $\mathbb{P}^{k-1}$ triangle, and the whole set constitutes the $\mathbb{P}^k$ triangle. With this construction, the $\mathbb{P}^k$ triangle is constituted of k layers of $\mathbb{P}^1$ sub-triangles: the $\mathbb{P}^1$ triangle is constituted of the layer 1, the $\mathbb{P}^2$ triangle is constituted of layers 1 and 2, and so on until the $\mathbb{P}^k$ triangle. Each layer $i, i = 1, ..., k$ is of length $2i - 1$.
With this numbering, we can now derive the general subdivision formula in the two dimensional case.
We first need the following preliminary results:

$$\begin{aligned}
&\int_K \nabla B_i = \frac{2}{k+1} \int_K \nabla \lambda_i, \ \ i = 1, 2, 3 \\
&\int_K \nabla B_i = \frac{2}{k+1} \Big(\int_K \nabla \lambda_{i_1} + \int_K \nabla \lambda_{i_2} \Big), \ \ i = 4, ..., 3k \\
&\int_K \nabla B_i = \frac{2}{k+1} \Big(\int_K \nabla \lambda_1 + \int_K \nabla \lambda_2 + \int_K \nabla \lambda_3 \Big) = 0, \\
&\quad i = 3k+1, ..., \frac{(k+1)(k+2)}{2}
\end{aligned} \tag{2.46}$$

where i_1 and i_2 are the vertices of the segment that contains the node i.
The results above are proved with the following formula (called "formule magique"[25]):

$$\int_K \lambda_1^{i_1} \ \lambda_2^{i_2} \ \lambda_3^{i_3} = 2 \ \text{mes}(K) \frac{i_1! \ i_2! \ i_3!}{(i_1 + i_2 + i_3 + 2)!}. \tag{2.47}$$

We can now present the subdivision formula for the k-th order in dimension two.

Proposition 1. In dimension two, we have the following formula, for $k \geq 1$:

$$\int_K \operatorname{div} \mathbf{f}^B dx = \frac{2}{k+1} \sum_{i=1}^{k} \Big(\int_{K_{(i-1)^2+1}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{(i-1)^2+3}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + ... + \int_{K_{i^2-2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{i^2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx \Big). \tag{2.48}$$

Proof. We denote by $\mathbf{n}_i$ the integral $\int_K \nabla \lambda_i \, dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned}
\int_K \operatorname{div} \mathbf{f}^B dx &= \sum_{j=1}^{\frac{(k+1)(k+2)}{2}} \mathbf{f}_j \cdot \int_K \nabla B_j \\
&= \frac{2}{k+1} \Big(\sum_{j=1}^{3} \mathbf{f}_j \cdot \int_K \nabla \lambda_j \\
&\quad + \sum_{j=4}^{3k} \mathbf{f}_j \cdot \big(\int_K \nabla \lambda_{j_1} + \int_K \nabla \lambda_{j_2} \big) \\
&\quad + \sum_{j=3k+1}^{\frac{(k+1)(k+2)}{2}} \mathbf{f}_j \cdot \big(\int_K \nabla \lambda_1 + \int_K \nabla \lambda_2 + \int_K \nabla \lambda_3 \big) \Big) \\
&= \frac{2}{k+1} \Big(\mathbf{f}_1 \cdot \mathbf{n}_1 + \mathbf{f}_2 \cdot \mathbf{n}_2 + \mathbf{f}_3 \cdot \mathbf{n}_3 \\
&\quad + \mathbf{f}_4 \cdot \big(\mathbf{n}_{4_1} + \mathbf{n}_{4_2} \big) + ... + \mathbf{f}_{3k} \cdot \big(\mathbf{n}_{3k_1} + \mathbf{n}_{3k_2} \big) \\
&\quad + \mathbf{f}_{3k+1} \cdot \big(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 \big) + ... + \mathbf{f}_{\frac{(k+1)(k+2)}{2}} \cdot \big(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 \big) \Big) \\
&= \frac{2}{k+1} \sum_{i=1}^{k} \Big(\int_{K_{(i-1)^2+1}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{(i-1)^2+3}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + ... \\
&\quad + \int_{K_{i^2-2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{i^2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx \Big).
\end{aligned} \tag{2.49}$$

□

We precise now the notations used in the proof. The indices in $\mathbf{n}_{j_1}$ and $\mathbf{n}_{j_2}$ represent the vertices of the segment containing the node j. The triangles K_l, $l = 1, ..., k^2$ are the $\mathbb{P}^1$ sub-triangles of the triangle K (see figure 2.5). For a given l, the vertices of K_l are denoted by l_1, l_2, l_3 and the function $\tilde{\mathbf{f}}^{(1)}$ is defined by:

$$\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_{l_1} \lambda_1 + \mathbf{f}_{l_2} \lambda_2 + \mathbf{f}_{l_3} \lambda_3 \ . \tag{2.50}$$

This definition depends on the ordering of the vertices of K_l. They are ordered in the reference triangle as in figure 2.4. The quantities $\mathbf{f}_j$ are not the values of $\mathbf{f}$ at the nodes j, they are considered as the values of $\mathbf{f}$ at the Bézier control points. Practically, they are computed from the definition of the Bézier basis functions (2.10) and by solving a linear system.

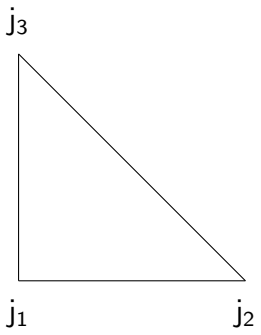


Figure 2.4: Ordering of vertices for relation (2.50).

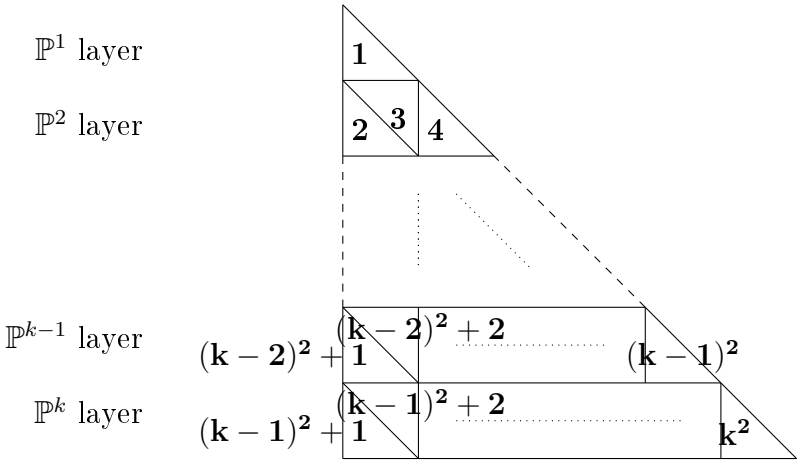


Figure 2.5: Numbering of $\mathbb{P}^1$ sub-triangles K_i in $\mathbb{P}^k$ triangle K.

The three dimensional case

The construction is similar to the two dimensional case, but slightly more complicated. The $\mathbb{P}^k$ tetrahedron is iteratively constructed by layers constituted of $\mathbb{P}^1$ tetrahedrons. Each i-layer, $i = 1, ..., k$ has a bottom face constituted of i^2 $\mathbb{P}^1$ triangles. The iterative layers and their bottom faces are described in figure 2.6. Let us now consider the bottom face of the layer i. Its triangles are numbered from the top to the bottom, like in the two dimensional case, as shown in figure 2.7. Each triangle is a face of a $\mathbb{P}^1$ tetrahedron of the layer i, and we number this tetrahedron with the number of such triangle. With this numbering, we can obtain a general three dimensional subdivision formula. We use again the Bézier basis functions (2.10) like in the two dimensional case, and we have the following preliminary results:

$$\begin{aligned}
\int_K \nabla B_i &= \frac{3}{\frac{(k+1)(k+2)}{2}} \int_K \nabla \lambda_i, \ i \in S_v \\
\int_K \nabla B_i &= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\int_K \nabla \lambda_{i_1} + \int_K \nabla \lambda_{i_2} \Big), \ i \in S_e \\
\int_K \nabla B_i &= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\int_K \nabla \lambda_{i_1} + \int_K \nabla \lambda_{i_2} + \int_K \nabla \lambda_{i_3} \Big), \ i \in S_f \\
\int_K \nabla B_i &= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\int_K \nabla \lambda_1 + \int_K \nabla \lambda_2 + \int_K \nabla \lambda_3 + \int_K \nabla \lambda_4 \Big) = 0, \ i \in S_t
\end{aligned} \tag{2.51}$$

where S_v is the set of vertices of K,
S_e is the set nodes situated on the edges of K, excepted the vertices (i_1 and i_2 are the vertices of the edge that contains the node i),
S_f is the set nodes situated on the faces of K, excepted those on the edges (i_1, i_2 and i_3 are the vertices of the face containing the node i),
S_t is the set of nodes situated strictly inside K,
and where B_i is the k^{th}-order Bézier basis function associated to the node i.
The results above are proved again with the "formule magique", here given in three dimensions:

$$\int_K \lambda_1^{i_1} \, \lambda_2^{i_2} \, \lambda_3^{i_3} \, \lambda_4^{i_4} = 6 \ \text{mes}(K) \frac{i_1! \ i_2! \ i_3! \ i_4!}{(i_1 + i_2 + i_3 + i_4 + 3)!}. \tag{2.52}$$

With these results we can now present the subdivision formula for the k-th order in dimension three.

Proposition 2. In dimension three, we have the following formula, for $k \geq 1$:

$$\int_K \operatorname{div} \mathbf{f}^B dx = \frac{3}{\frac{(k+1)(k+2)}{2}} \sum_{i=1}^{k} \sum_{i'=1}^{i} \Big(\int_{K_{(i'-1)^2+1}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{(i'-1)^2+3}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + ... + \int_{K_{i'^2-2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{i'^2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx \Big). \tag{2.53}$$

Proof. We denote by $\mathbf{n}_i$ the integral $\int_K \nabla \lambda_i \, dx$, $i = 1, 2, 3$ and we have:

$$\begin{aligned}
\int_K \operatorname{div} \mathbf{f}^B dx &= \sum_{j=1}^{\frac{(k+1)(k+2)(k+3)}{6}} \mathbf{f}_j \cdot \int_K \nabla B_j \\
&= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\sum_{j \in S_v} \mathbf{f}_j \cdot \int_K \nabla \lambda_j \\
&\qquad + \sum_{j \in S_e} \mathbf{f}_j \cdot \big(\int_K \nabla \lambda_{j_1} + \int_K \nabla \lambda_{j_2} \big) \\
&\qquad + \sum_{j \in S_f} \mathbf{f}_j \cdot \big(\int_K \nabla \lambda_{j_1} + \int_K \nabla \lambda_{j_2} + \int_K \nabla \lambda_{j_3} \big) \\
&\qquad + \sum_{j \in S_t} \mathbf{f}_j \cdot \big(\int_K \nabla \lambda_1 + \int_K \nabla \lambda_2 + \int_K \nabla \lambda_3 + \int_K \nabla \lambda_4 \big) \Big) \\
&= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\sum_{j \in S_v} \mathbf{f}_j \cdot \mathbf{n}_j \\
&\qquad + \sum_{j \in S_e} \mathbf{f}_j \cdot \big(\mathbf{n}_{j_1} + \mathbf{n}_{j_2} \big) \\
&\qquad + \sum_{j \in S_f} \mathbf{f}_j \cdot \big(\mathbf{n}_{j_1} + \mathbf{n}_{j_2} + \mathbf{n}_{j_3} \big) \\
&\qquad + \sum_{j \in S_t} \mathbf{f}_j \cdot \big(\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4 \big) \Big) \\
&= \frac{3}{\frac{(k+1)(k+2)}{2}} \Big(\mathbf{f}_{1_{1_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{1_{1_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{1_{1_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{1_{1_4}} \cdot \mathbf{n}_4 \quad \text{(lay. 1)} \\
&+ \mathbf{f}_{2_{1_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{2_{1_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{2_{1_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{2_{1_4}} \cdot \mathbf{n}_4 \quad \text{(lay. 2, tet. 1, sublay. 1)} \\
&+ \mathbf{f}_{2_{2_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{2_{2_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{2_{2_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{2_{2_4}} \cdot \mathbf{n}_4 \quad \text{(lay. 2, tet. 2, sublay. 2)} \\
&+ \mathbf{f}_{2_{4_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{2_{4_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{2_{4_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{2_{4_4}} \cdot \mathbf{n}_4 \quad \text{(lay. 2, tet. 4, sublay. 2)} \\
&+ ... \\
&+ \mathbf{f}_{k_{1_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{k_{1_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{k_{1_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{k_{1_4}} \cdot \mathbf{n}_4 \quad \text{(lay. } k\text{, tet. 1, sublay. 1)} \\
&+ \mathbf{f}_{k_{2_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{k_{2_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{k_{2_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{k_{2_4}} \cdot \mathbf{n}_4 \quad \text{(lay. } k\text{, tet. 2, sublay. 2)}
\end{aligned}$$

$$
\begin{aligned}
&+ \mathbf{f}_{k_{4_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{k_{4_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{k_{4_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{k_{4_4}} \cdot \mathbf{n}_4 \quad \text{(lay. } k\text{, tet. 4, sublay. 2)} \\
&+ ... \\
&+ \mathbf{f}_{k_{(k-1)^2+1_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{k_{(k-1)^2+1_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{k_{(k-1)^2+1_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{k_{(k-1)^2+1_4}} \cdot \mathbf{n}_4 \text{ (lay. } k\text{, tet. } (k-1)^2 \\
&+ ... \\
&+ \mathbf{f}_{k_{k^2{}_1}} \cdot \mathbf{n}_1 + \mathbf{f}_{k_{k^2{}_2}} \cdot \mathbf{n}_2 + \mathbf{f}_{k_{k^2{}_3}} \cdot \mathbf{n}_3 + \mathbf{f}_{k_{k^2{}_4}} \cdot \mathbf{n}_4 \quad \text{(lay. } k\text{, tet. } k^2\text{, sublay. } k) \Big) \\
&= \frac{3}{\frac{(k+1)(k+2)}{2}} \sum_{i=1}^{k} \sum_{i'=1}^{i} \Big(\int_{K_{(i'-1)^2+1}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{(i'-1)^2+3}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + ... \\
&\qquad + \int_{K_{i'^2-2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx + \int_{K_{i'^2}} \operatorname{div} \tilde{\mathbf{f}}^{(1)} dx \Big).
\end{aligned}
$$

□

Let us precise the notations used in the proof. We denote by $\mathbf{f}_j$ the value of the flux at the global node j, and by $\mathbf{f}_{i_{\alpha_\beta}}$ the flux at the node β of the tetrahedron α of the layer i. We precise the sublayer, it indicates in which layer of the bottom face of triangles the tetrahedron is taken (see figure 2.7). It corresponds to the index i'. The indices in $\mathbf{n}_{j_1}$, $\mathbf{n}_{j_2}$ (for $j \in S_e$) represent the vertices of the edge containing the node j and in $\mathbf{n}_{j_1}$, $\mathbf{n}_{j_2}$, $\mathbf{n}_{j_3}$ (for j $\in S_f$) they represent the vertices of the face containing the node j. For a given layer i, $i = 1, ..., k$, the tetrahedrons K_l are $\mathbb{P}^1$ sub-tetrahedrons of the tetrahedron K in the layer i. The number l, with $l = (i-1)^2, ..., i^2$, is local to the layer i and coincides with the number of the triangle l of the bottom face of the layer i which is a face of K_l (see figure 2.6 and figure 2.7). With this numbering, we have a general formula in the three dimensional case that is similar to the two dimensional case. For a given l, the vertices of K_l are denoted by l_1, l_2, l_3, l_4 and the function $\tilde{\mathbf{f}}^{(1)}$ is defined by:

$$
\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_{l_1}\lambda_1 + \mathbf{f}_{l_2}\lambda_2 + \mathbf{f}_{l_3}\lambda_3 + \mathbf{f}_{l_4}\lambda_4 \ . \tag{2.54}
$$

This definition depends on the ordering of the vertices of K_l. They are ordered in the reference tetrahedron as in figure 2.8. Like in the two dimensional case, the quantities $\mathbf{f}_j$ are not the values of $\mathbf{f}$ at the nodes j, they are considered as the values of $\mathbf{f}$ at the Bézier control points and in practice are computed from the definition of the Bézier basis functions (2.10) and by solving a linear system.

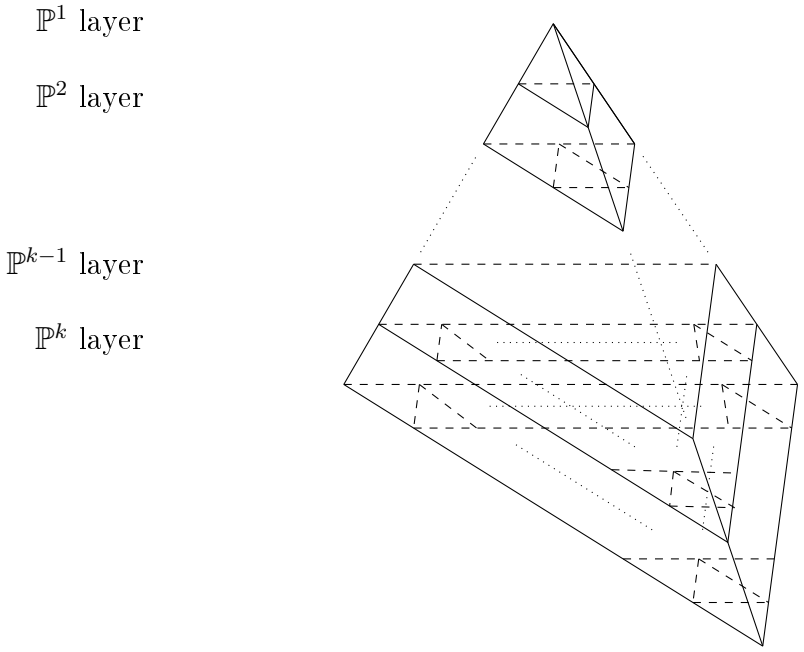


Figure 2.6: Subdivided $\mathbb{P}^k$ tetrahedron K: layers and their bottom faces.

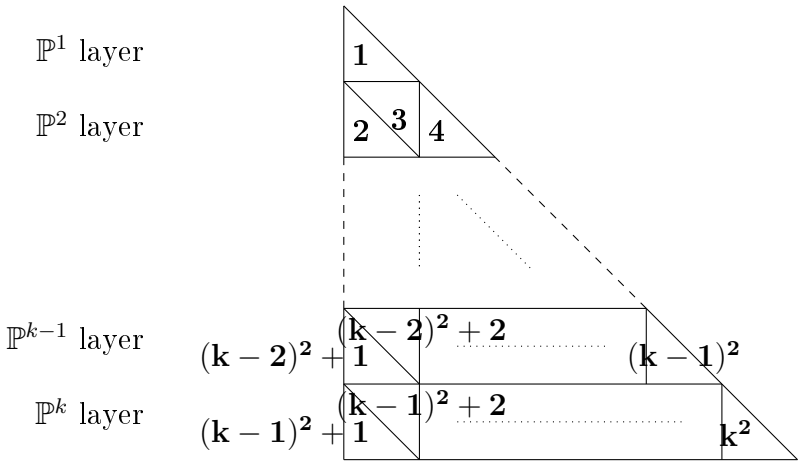


Figure 2.7: Numbering of $\mathbb{P}^1$ sub-triangles constituting the bottom face of the $\mathbb{P}^k$ layer of tetrahedron K.

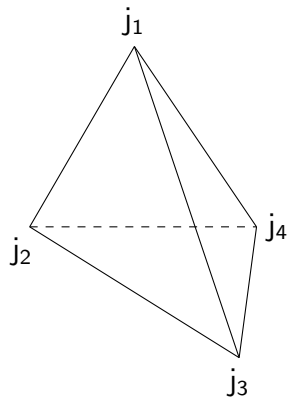


Figure 2.8: Ordering of vertices

2.3.4 Definition of the nodal residuals

We denote here by S the set of $\mathbb{P}^1$ sub-elements of a given element K. The term element represents a triangle in dimension two or a tetrahedron in dimension three as above. For a given element K subdivided into smaller elements indexed by $\xi, \xi \in S$ (for example $\xi = K_1, ..., K4$ in the case of quadratic interpolation in dimension two), we have, as we have proved, a relation between the residual Φ^K and the residuals Φ^ξ of the sub-elements of K. Once the residuals Φ^ξ are computed, the next step is to compute the nodal residuals Φ^ξ_σ, as described in section 2.2.2. Then, the nodal residuals Φ^ξ_σ need to be modified in order to satisfy the conservation relation (2.19), as we show now. We denote by $\gamma_\xi, \xi \in S$ the coefficients given by the formulas of section 2.3.3.

Proposition 3. If the nodal residuals $\Phi^\xi_\sigma, \sigma \in \Sigma_\xi, \xi \in S$ satisfy the conservation relation (2.19), then the nodal residuals defined by

$$\Phi^K_\sigma := \sum_{\xi \in S, \sigma \in \Sigma_\xi} \gamma_\xi \Phi^\xi_\sigma \tag{2.55}$$

satisfy the conservation relation (2.19).

Proof. With the notations of section 2.3, $\mathbf{f}^{(k)}$ represents the k-th order interpolant of the flux $\mathbf{f}$. We assume that

$$\forall \xi \in S, \sum_{\sigma \in \Sigma_\xi} \Phi^\xi_\sigma = \int_{\partial \xi} \mathbf{f}^{(1)} \cdot \mathbf{n}$$

and that

$$\Phi^K_\sigma = \sum_{\xi \in S, \sigma \in \Sigma_\xi} \gamma_\xi \Phi^\xi_\sigma.$$

We then have

$$\begin{aligned}\sum_{\sigma\in\Sigma_K}\Phi_\sigma^K &= \sum_{\sigma\in\Sigma_K}\left(\sum_{\xi\in S,\sigma\in\Sigma_\xi}\gamma_\xi\Phi_\sigma^\xi\right)\\ &= \sum_{\xi\in S}\gamma_\xi\left(\sum_{\sigma\in\Sigma_\xi}\Phi_\sigma^\xi\right)\\ &= \sum_{\xi\in S}\gamma_\xi\int_{\partial\xi}\mathbf{f}^{(1)}\cdot\mathbf{n}\\ &= \int_{\partial K}\mathbf{f}^{(k)}\cdot\mathbf{n}\,dx.\end{aligned}$$

□

We remark that it is possible to use any scheme inside the elements $\xi\in S$ as long as they satisfy the conservation relation (2.19) and the nodal residuals in K are defined by relation (2.55). Under the assumption of the Lax-Wendroff theorem, the new scheme with arbitrary mixed $\mathbb{P}^1$ and $\mathbb{P}^k$ elements will also be convergent to a weak solution of the problem as the conservation property (2.19) is still satisfied (see [6]). We follow the same reasoning for the boundary residuals.

2.3.5 Practical implementation

The nodal residuals are modified according to the definition given in section 2.3.4. We describe now the consequences on the practical implementation of the Residual Distribution scheme.

Computation of the nodal residuals

As the computation of the nodal residuals is made inside each sub-element ξ of a divided element K, it is convenient to introduce $\Phi_\sigma^{K,\xi}$ the contribution of the node σ in triangle K brought by the sub-element ξ, as it is the quantity that is actually computed. It is defined by

$$\Phi_\sigma^{K,\xi} := \gamma_\xi\,\Phi_\sigma^\xi. \tag{2.56}$$

Computation of the Jacobian matrix of the implicit scheme

The system of equations (2.21) is solved with an implicit scheme detailed in Appendix 2.A. To compute the Jacobian matrix, for example in the case of the Lax-Wendroff scheme (2.22), we need to differentiate the two terms of (2.22):

$$\frac{\partial\Phi_\sigma^\xi(\mathbf{u}_h)}{\partial\mathbf{u}_j} = \frac{1}{N_{dof}^\xi}\frac{\partial}{\partial\mathbf{u}_j}\Big(\,\Phi^\xi(\mathbf{u}_h) + \int_\xi A\cdot\nabla\varphi_\sigma\,\Xi\,A\cdot\nabla\mathbf{u}_h dx\,\Big) \tag{2.57}$$

and so, according to (2.55), we have, with the same notations as in section 2.3.5:

$$\frac{\partial \Phi_\sigma^{K,\xi}(\mathbf{u}_h)}{\partial \mathbf{u}_j} = \gamma_\xi \frac{\partial \Phi_\sigma^{\xi}(\mathbf{u}_h)}{\partial \mathbf{u}_j}. \tag{2.58}$$

Boundary conditions

The nodal residuals of boundary faces (here we use the term face for either the edge of a triangle in dimension two or the face of a tetrahedron in dimension three) are computed by using the formula (2.37). If we denote by Γ the face of a subdivided element K lying on the boundary of Ω, Γ is therefore subdivided into sub-faces. Let ς be such a sub-face. From the relation (2.55), we see that in order to be consistent with relation (2.37) in the case of subdivision, we need to multiply the nodal residuals of the subdivided boundary by the subdivision coefficient of the element containing this sub-divided boundary face, and so the contribution of the node σ in the face Γ brought by the sub-face ξ writes

$$\Phi_\sigma^{\Gamma,\varsigma} = \gamma_\xi \, \Phi_\sigma^{\varsigma} \tag{2.59}$$

and thus the contribution to the global Jacobian writes

$$\frac{\partial \Phi_\sigma^{\Gamma,\varsigma}(\mathbf{u}_h)}{\partial \mathbf{u}_j} = \gamma_\xi \frac{\partial \Phi_\sigma^{\varsigma}(\mathbf{u}_h)}{\partial \mathbf{u}_j}. \tag{2.60}$$

Choice of quadrature formulas

As explained in section 2.2.5, the relation (2.40) is verified if we use the same quadrature points at the interfaces between elements. In the case of an interface between two elements of the same degree, we simply use the same quadrature formula, and so the requirements of relation (2.40) are automatically satisfied. For an interface between a subdivided element and a non-subdivided one, we use quadrature formulas such that all the quadrature points at the interface physically coincide.

2.4 Numerical results

We present now some numerical results for different speed flows in dimension two and three that illustrate the theoretical results exposed above. Even if the formulas presented in section 2.3.3 make it possible to use any polynomial order, for simplicity the test cases presented in this chapter use $\mathbb{P}^1$ and $\mathbb{P}^2$ elements. As explained in section 2.2.3, for subsonic flows we use the Lax-Wendroff scheme (2.22) and for transonic and faster flows we use the Rusanov scheme (2.36). Our approach can be described as follows. In all test cases except the subsonic test case, the mesh is mostly made of $\mathbb{P}^2$ elements except

in the shock zone where $\mathbb{P}^1$ elements are used. The shock zone is located by a shock detector (see [10]) based here on the variation of pressure inside each element of the mesh:

$$\theta_K = \max_{\sigma \in \Sigma_K} \left(\max_{K', \sigma \in \Sigma_{K'}} \frac{|\max_{\nu \in \Sigma_{K'}} p_\nu - \min_{\nu \in \Sigma_{K'}} p_\nu|}{|\max_{\nu \in \Sigma_{K'}} p_\nu| + |\min_{\nu \in \Sigma_{K'}} p_\nu| + \epsilon} \right) \tag{2.61}$$

where p_ν is the pressure at the node ν, K, K' are elements of the mesh $\mathcal{T}_h$ and ϵ is the machine epsilon (in our implementation, $\epsilon = 1e-16$). The shock detector is used as follows. We set a threshold θ which depends on the test case, we start with a mesh that contains only subdivided $\mathbb{P}^1$ elements, then after a short number of iterations (which depends on the test case too), the values θ_K are computed for each element K. If the value θ_K is above the threshold θ, then the element remains subdivided with $\mathbb{P}^1$ elements, otherwise, the element becomes a $\mathbb{P}^2$ element and is not subdivided. As a result, we use $\mathbb{P}^1$ elements in the shock zone, and $\mathbb{P}^2$ elements everywhere else.

2.4.1 Subsonic flow

We show here, just for theoretical purposes, how our method behaves with the mesh of a NACA0012 wing profile constituted of randomly subdivided elements. The inflow condition is Mach=0.5, the pressure is $P_{inlet} = 0.7$ and half of the elements of the mesh are randomly selected and subdivided into $\mathbb{P}^1$ elements. As the flow is subsonic, we use the Lax-Wendroff scheme (2.22). The mesh is shown in figure 2.9.

We can see in figure 2.10 a comparison of the convergence curves when $\mathbb{P}^1$, $\mathbb{P}^2$ and mixed elements ($\mathbb{P}^2$ elements and subdivided $\mathbb{P}^1$ elements) are used. We remark that with the same level of convergence for the three types of finite elements, the convergence speed with subdivided elements is as expected between those obtained with $\mathbb{P}^1$ and $\mathbb{P}^2$ elements.

As stated above, for this test case the elements are arbitrarily subdivided and we do not take advantage of p-adaptation to improve the accuracy of the solution. In the next test cases, the elements will be subdivided according to the properties of the solution.

2.4.2 Transonic flow

We test our method on a Naca0012 wing profile with Mach=0.8, a pressure of 0.71 and an angle of attack of 1.25 degrees. For this problem, we use the Rusanov scheme (2.36), more adapted for transonic flows than the Lax-Wendroff scheme, as explained in section 2.2.3. As small law order elements capture irregular solutions better than high order elements, which to the contrary approximate smooth solutions better than low order elements [23], we use the

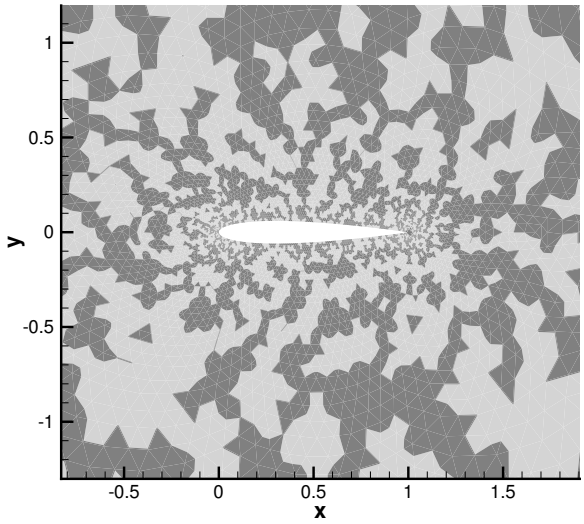


Figure 2.9: Mesh for test case 2.4.1 with elements randomly subdivided: elements in dark zones are subdivided, the others are not subdivided.

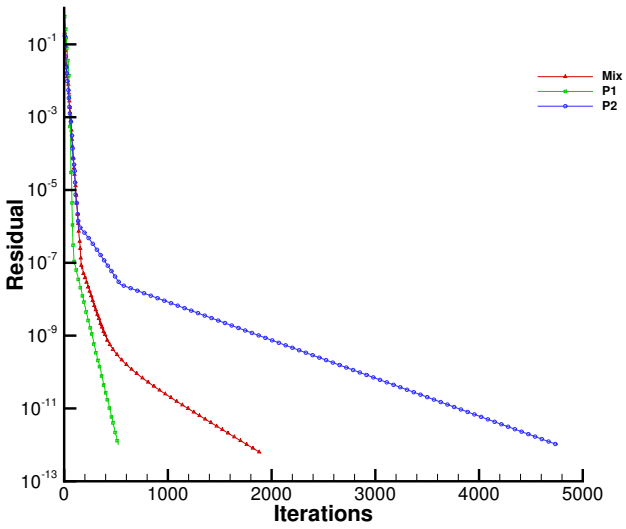


Figure 2.10: Convergence of residuals for test case 2.4.1.

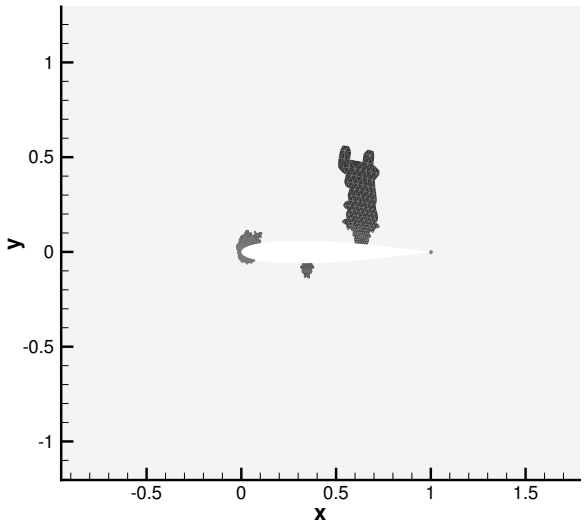


Figure 2.11: Shock zone for test case 2.4.2: subdivided elements are dark-colored.

shock detector (2.61) that allows to use $\mathbb{P}^1$ elements only where the shock is detected and $\mathbb{P}^2$ elements otherwise. The mesh is shown in figure 2.11, the subdivided elements are dark-colored and correspond to the elements containing a strong variation of pressure. The convergence curve obtained with p-adaptation, figure 2.12, shows a jump of the residual due to the switch from $\mathbb{P}^1$ elements to $\mathbb{P}^2$ elements everywhere except in the shock zone. After the jump, we observe the convergence of the residual. For this test case θ was set to 1.5, and the switch was set after 300 iterations. We remark from figure 2.13, that the position of the shock obtained with mixed elements is in very good agreement with the position predicted using classical $\mathbb{P}^2$ or $\mathbb{P}^1$ elements. This is important for the validation of our method and proves that we are here consistent with the results obtained with classical $\mathbb{P}^1$ and $\mathbb{P}^2$ elements.
In addition, we make the following interesting observation. As we use smaller $\mathbb{P}^1$ elements in the discontinuous zone where they are better suited to capture a discontinuity than coarser $\mathbb{P}^2$ elements, we obtain a better representation of the shock, as with the $\mathbb{P}^1$ mesh. This observation is confirmed if we compare, in figure 2.14, the solutions obtained with p-adaptation to the solution obtained with a $\mathbb{P}^2$ mesh. We can see that the shock is better represented with the p-adaptive solution.

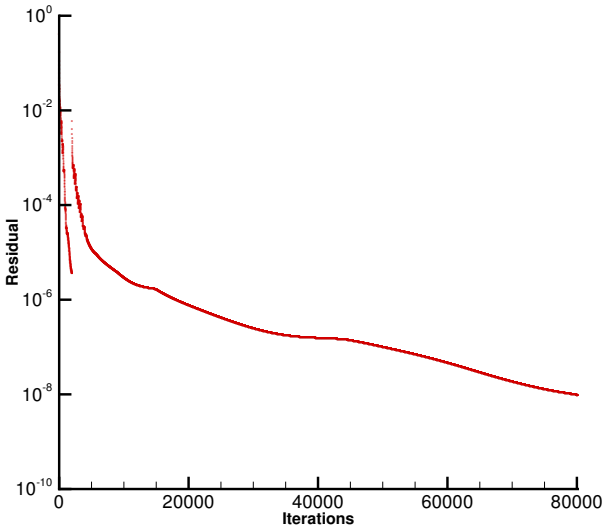


Figure 2.12: Convergence of residual for test case 2.4.2 with mixed elements.

2.4.3 Supersonic flow

Now we present a numerical test with a higher speed, Mach=3. Because of the higher speed used, such test cases can be more difficult to run. The application of p-adaptation, even to a little number of elements (only in the shock zone), allows convergence of the residual and proves to be an efficient approach to simulate such phenomenon. Since the number of sub-divided elements is small the method remains mostly $\mathbb{P}^2$ based.

The following parameters are set: the mesh is made of 3749 vertices and contains a sphere of diameter 1, centered in 0, which is moving at Mach=3.0. We divide the boundary conditions into four sub-boundaries as shown in figure 2.15, and we detail in the following the conditions applied to each of these boundaries:

- on the sphere (boundary 1) inside the domain, we impose a slipping wall boundary condition,
- in front of the sphere (boundary 2, the half circle on the left of the domain), we impose Dirichlet boundary conditions with $(\rho, u, v, p) = (1,\ 3,\ 0,\ 1.4)$,
- on the upper and lower horizontal lines (boundary 3), we impose a slipping wall boundary condition,
- behind the sphere (boundary 4, the vertical line at the right of the domain), we use a Steger-Warming exit boundary condition with $(\rho, u, v, p) =$

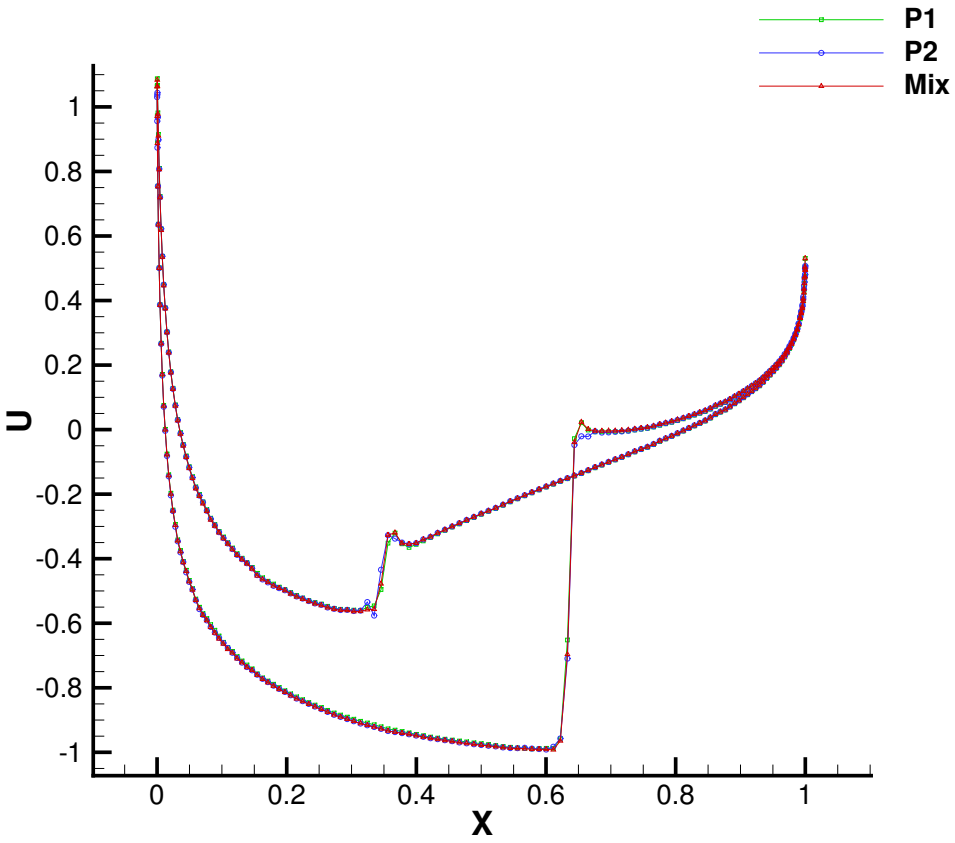
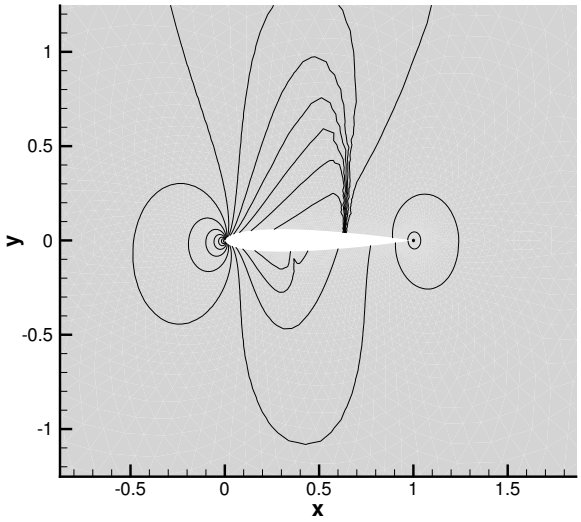
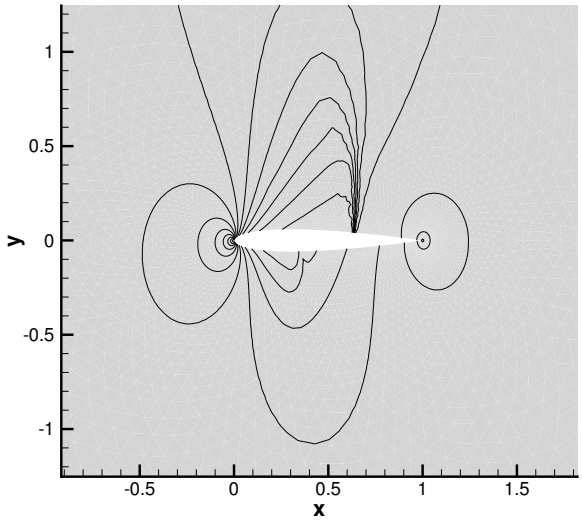


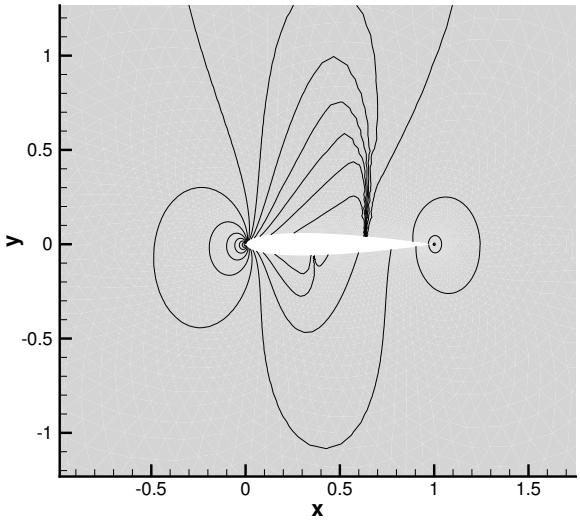
Figure 2.13: Comparison of pressure coefficients obtained with $\mathbb{P}^1$, $\mathbb{P}^2$ and mixed elements.



(a) Pressure ($\mathbb{P}^1$ elements)



(b) Pressure ($\mathbb{P}^2$ elements)



(c) Pressure (mixed elements)

Figure 2.14: Comparison of solutions for test case 2.4.2 between $\mathbb{P}^2$ and mixed elements:

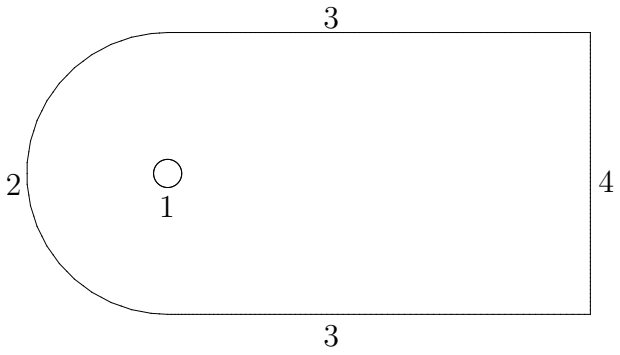


Figure 2.15: Boundaries for test case 2.4.3.

(1, 0.8, 0, 0.3).

As initial conditions we set a discontinuity line at $x = 0.435$, with $(\rho, u, v, p) = (1,\ 0,\ 0,\ 1.4)$ on the left of the discontinuity and $(\rho, u, v, p) = (1,\ 3,\ 0,\ 1.4)$ on the right.

We use the same method as before with the shock detector (2.61) and obtain the mesh shown in figure 2.16. The threshold is set at $\theta = 3$ and the switch is set at 300 iterations. Again, we notice in figure 2.17 a jump in the residual due to the change from a $\mathbb{P}^1$-only scheme to a mixed $\mathbb{P}^1$-$\mathbb{P}^2$ scheme after the activation of the shock detector.

The isolines of the Mach number, the pressure and the density are shown in respectively figures 2.18, 2.19 and 2.20. As of today, we have not been able to make the classical $\mathbb{P}^2$ scheme converge for this test case. With p-adaptation we obtain a high-order solution that physically agrees with the solution obtained with a classical $\mathbb{P}^1$ scheme.

2.4.4 Hypersonic three dimensional flow

We present now a numerical test case with a hypersonic speed (Mach=8) in dimension three. Like in dimension two, the application of p-adaptation to a little number of elements in the shock zone (the method remaining mostly $\mathbb{P}^2$ based), proves practically to be efficient as it allows the convergence of the residual and gives a solution that looks physically admissible.
The following parameters are set: a sphere of diameter two is centered in 0 and is moving at the speed of Mach=8. The boundary conditions are divided into four sub-boundaries, as shown in figure 2.21. On the sphere inside the domain (boundary 1), we impose a slipping wall boundary condition, in the left face of the domain (boundary 2), we impose a Steger-Warming entry boundary condition, with $(\rho, u, v, w, p) = (8.0\ ,\ 8.25\ ,\ 0.0\ ,\ 0.0\ ,\ 116.5)$, in the right face of the domain (boundary 3), we impose a Steger-Warming exit boundary condition,

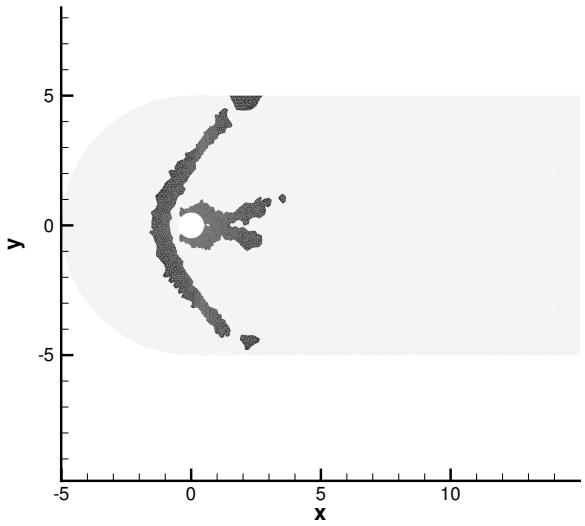


Figure 2.16: Shock zone for test case 2.4.3: subdivided elements are dark-colored.

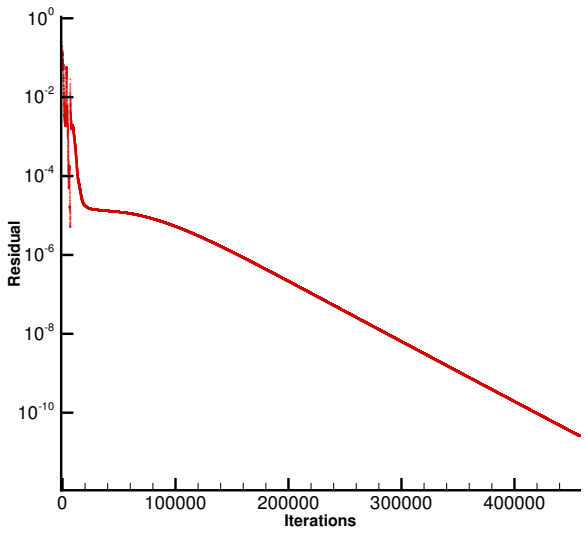


Figure 2.17: Convergence of residual for test case 2.4.3.

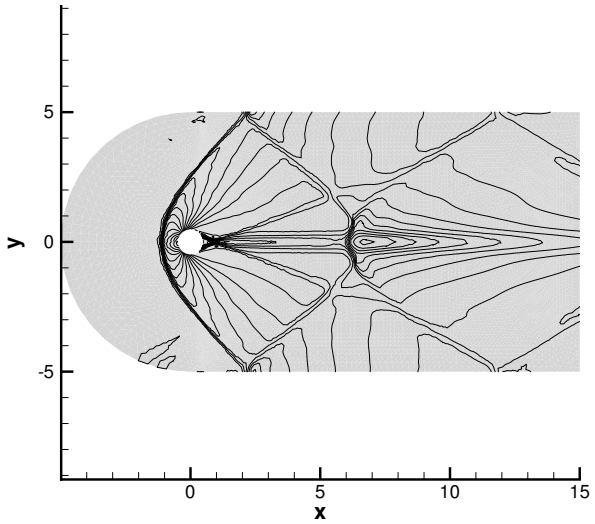


Figure 2.18: Mach number isolines for test case 2.4.3.

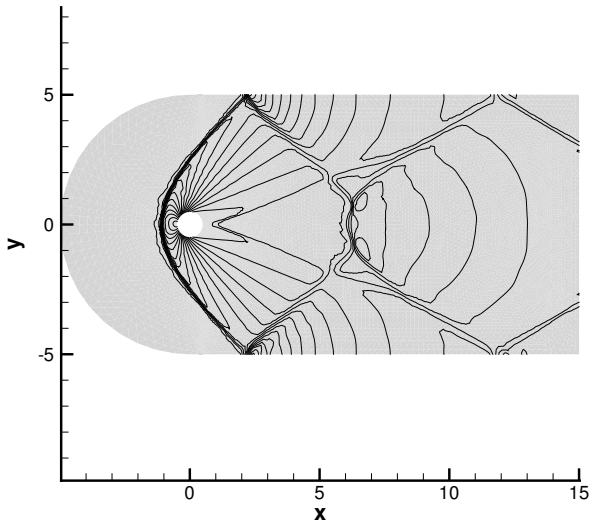


Figure 2.19: Pressure isolines for test case 2.4.3.

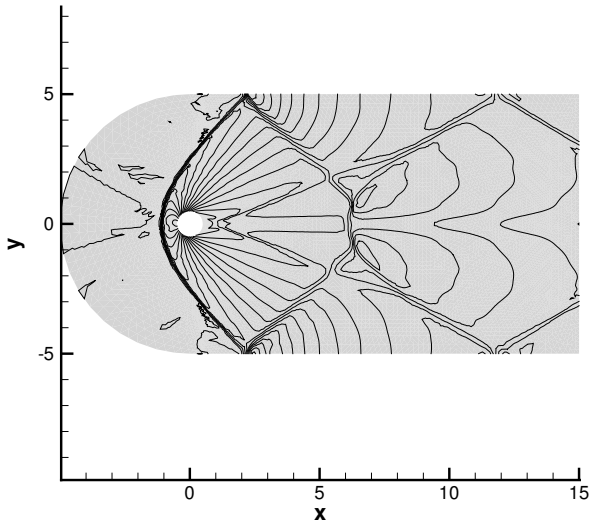


Figure 2.20: Density isolines for test case 2.4.3.

with $(\rho, u, v, w, p) = (1.4\ ,\ 0.0\ ,\ 0.0\ ,\ 0.0\ ,\ 1.0)$, and on the other faces of the boundary (boundary 4), we impose a slipping wall boundary condition. As an initial condition we set a vertical plan of discontinuity at $x = 0.09$, with $(\rho, u, v, w, p) = (8.0\ ,\ 8.25\ ,\ 0.0\ ,\ 0.0\ ,\ 116.5)$ at the left of the discontinuity, and $(\rho, u, v, w, p) = (1.4\ ,\ 0.0\ ,\ 0.0\ ,\ 0.0\ ,\ 1.0)$ at the right of the discontinuity.

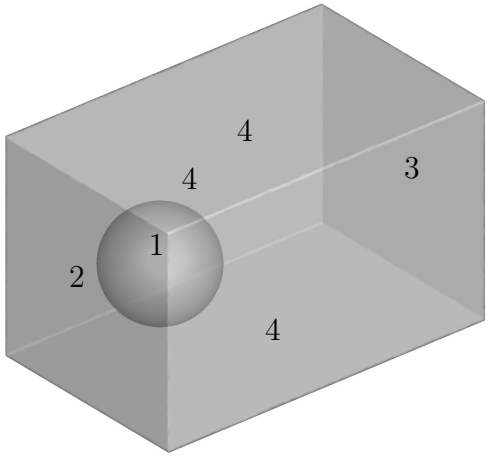


Figure 2.21: Boundaries for test case 2.4.4.

As shown in figure 2.22, the residual converges well, even with only a very small number of $\mathbb{P}^1$ elements in the shock zone (see figure 2.23). Like in test case 2.4.3, with here a threshold $\theta = 8$, we start with subdivided $\mathbb{P}^1$ elements everywhere. After 300 iterations, only elements in the shock zone remain $\mathbb{P}^1$, all the others become $\mathbb{P}^2$ elements. The converged solution (Mach number, pressure and density), is shown in respectively figures 2.24, 2.25 and 2.26.

2.5 Conclusion

We have described a way to use p-adaptation with continuous finite elements within the frame of residual distribution schemes. We have showed with general formulas that the method can be theoretically extended to arbitrary polynomial orders, and we have shown for complex problems modeled by the Euler equations that in practice, in the case of quadratic approximation in dimension two and three, the method is robust for subsonic, transonic, supersonic and hypersonic flows. Other applications can be envisaged and in particular, the extension of our method to the equations of Navier-Stokes coupled with the Penalization method will be the subject of chapter (3).

Acknowledgments

- This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02).

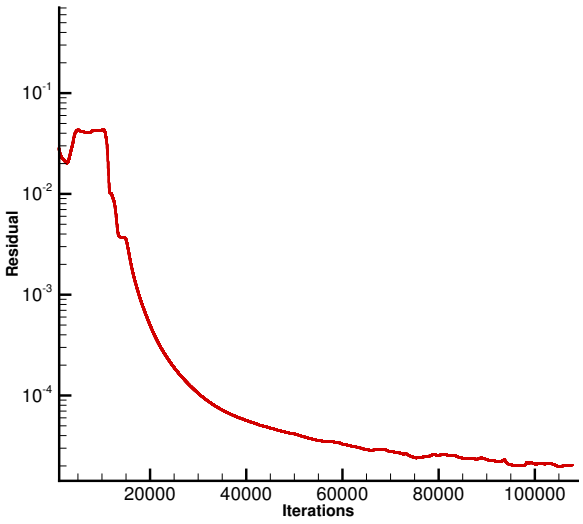


Figure 2.22: Convergence of residual for test case 2.4.4

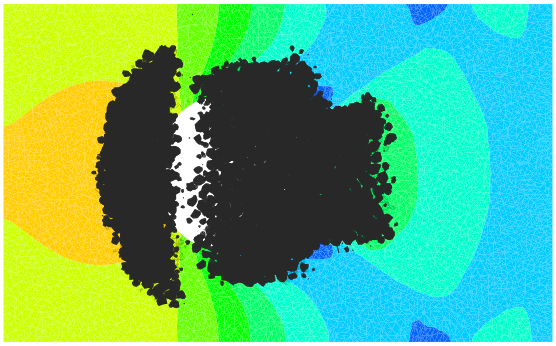


Figure 2.23: Shock zone for test case 2.4.4(with two dimensional slice cut at $y = 0$ of the pressure).

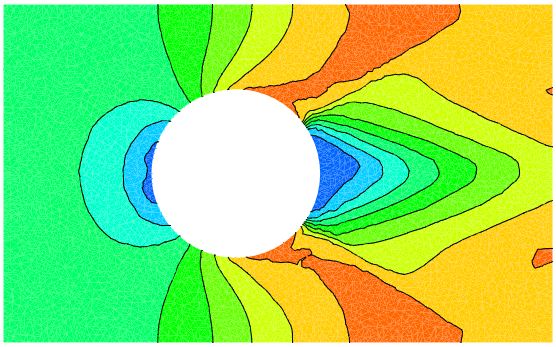


Figure 2.24: Mach number isolines for test case 2.4.4 (two dimensional slice cut at $y = 0$).

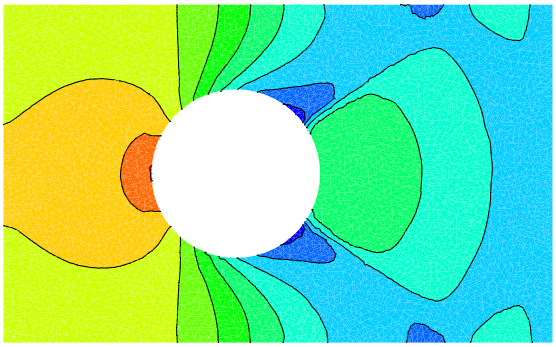


Figure 2.25: Pressure isolines for test case 2.4.4 (two dimensional slice cut at $y = 0$).

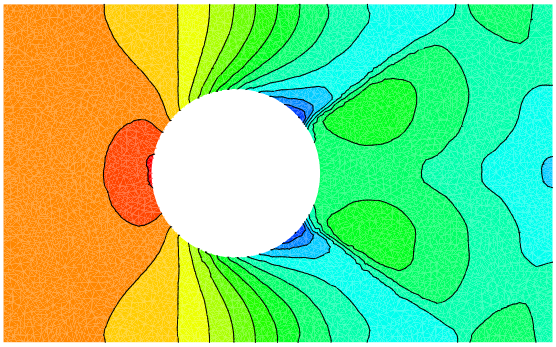


Figure 2.26: Density isolines for test case 2.4.4 (two dimensional slice cut at $y = 0$).

- Experiments presented in this chapter were carried out using the cluster Avakas and the PLAFRIM experimental testbed. PLAFRIM is developed under the Inria PlaFRIM development action with support from LABRI, IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS.

2.A Implicit numerical solver

We need to solve the system of equations (2.21), written in a compact way as:

$$R(\mathbf{u}_h) = 0. \tag{62}$$

This problem is first relaxed as:

$$\frac{d\mathbf{u}_h}{dt} = -R(\mathbf{u}_h). \tag{63}$$

To approximate this time derivative, we use the backward Euler formula, and the problem becomes:

$$\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t^n} = -R(\mathbf{u}_h^{n+1}), n = 0, 1, 2, ... \tag{64}$$

Thus, we have to solve a non linear problem at each time step n. We use for that a Newton method, that when applied to (62) reads:

$$\mathbf{u}_h^{k+1} = \mathbf{u}_h^k - J^{-1}R(\mathbf{u}_h^k), \quad k = 0, 1, 2, ... \tag{65}$$

where

$$J = \frac{\partial R(\mathbf{u}_h^k)}{\partial \mathbf{u}} \tag{66}$$

is the Jacobian of R. In practice, the Jacobian J is computed from a first order scheme with the stabilization term.

And so, with our Newton method applied to (64) , the linear system we have to solve at each time step n reads:

$$[\frac{1}{\Delta t^n} + J(\mathbf{u}_{h_k}^n)]\Delta\mathbf{u}_{h_k}^n = -R(\mathbf{u}_{h_k}^n) \tag{67}$$

$$\mathbf{u}_{h_{k+1}}^n = \mathbf{u}_{h_k}^n + \Delta\mathbf{u}_{h_k}^n, \quad k = 0, 1, 2... \tag{68}$$

In practice, for each time step n, we use only one iteration on k, which seems enough to reach convergence of the nodal residuals to the zero machine.

2.B Comparison with constrained approximation for the finite element method

In [18] the authors present a mesh adaptation method called "constrained approximation" in the frame of a Petrov-Galerkin finite-element method. The idea behind the method is to use a non regular mesh (non conformal and of various order) and to constrain the value of the discrete solution at the interface shared by elements of different size (h-adaptation) and different order (p-adaptation) to keep the solution continuous across the interface.

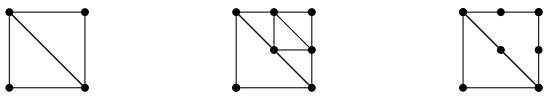


Figure 27: From left to right: initial mesh, h-adaptation and p-adaptation.

In figure 27, the authors give an example of h-refinement and p-refinement. In the h-refinement case, the two triangles share a common edge with two different sizes, which generates a hanging node in the middle of the edge. In the p-refinement case, the triangles are of two different polynomial orders ($\mathbb{P}1$ and $\mathbb{P}2$) and so a hanging node is generated in the middle of the edge. In both cases, the solution is constrained at the hanging node so that it remains continuous.

The authors use a hierarchical basis of shape functions [81]. For example, in dimension two this basis is constituted of functions associated with the vertices, the edges and the element and are called accordingly vertex, edge and bubble functions. They are built incrementally with respect to the desired order of the approximation, by using kernel functions in the definition of the edge and bubble functions. The main advantages of this basis over the classical Lagrange basis functions, is that the polynomial order of the approximation

can be changed easily (by adding or removing kernel functions) and in the case of p-adaptation, like in figure 27, the hanging node is naturally suppressed, by changing to one the polynomial order of the shape function associated with the edge at the interface in the $\mathbb{P}2$ element.

Once the constrains are established, the next step is to apply these constrains. The authors present a complete set of rules to impose the continuity of the solution in the case of h-adaptation and p-adaptation. The authors suggest to follow a rule called "1-irregularity" rule (only one hanging node between two elements) in order to avoid too complex situations, at least for h-adaptation.

We believe that this method is very different to the method we have proposed in this chapter. The first difference is that the generation of hanging nodes makes the mesh non conformal. In a RD scheme, the question of how this hanging node should be treated does not seem evident. The distribution schemes (like for example (2.22) and (2.36)) would probably have to be redefined to take into account the hanging nodes, so that the scheme remains conservative (which is an essential ingredient for the convergence) and still exhibit the same behavior (for example (non) diffusive or (non) oscillatory). Even if it is still possible to use a classical Lagrange basis, it is advised in [18] to use a hierarchical basis of functions, as such a basis is more suited for p-adaptation. So, the compatibility of a hierarchical basis function with RD schemes should be analyzed. Finally, there is the question as to whether the hanging nodes should be taken into account in the system of equations (2.21), and for the implementation of an implicit RD scheme, what the consequences would be for the computation of the Jacobian matrix. We are not saying that the constrained approximation method is not compatible with RD schemes (indeed, we have obtained some preliminary numerical results with a prototype of a RD scheme based upon the constrained approximation method, in the case of scalar equations with Lagrange elements and an explicit scheme). But the questions we have raised should first be carefully assessed before trying to make an RD scheme compatible with the adaptive method of [18]. To the contrary, the p-adaptive method for RD schemes we have proposed here is designed for RD schemes by construction and as such avoids all the problems evoked above.

Chapter 3

A hp-adaptive continuous Residual Distribution scheme for the penalized Navier-Stokes equations

This chapter consists of a second journal paper submitted.

In this chapter, a method that combines hp-adaptation and penalization within a Residual Distribution scheme is proposed. The IBM-LS-AUM method is an embedded boundary method that provides a simple and accurate treatment of the wall boundary conditions by the technique of penalization and anisotropic mesh adaptation. We present in this chapter an hp-adaptive scheme for the penalized Navier-Stokes equations in the frame of a continuous Residual Distribution scheme. The method can be considered as an evolution of the IBM-LS-AUM method to higher order elements and is based on the construction of a p-adaptive Residual Distribution scheme that was recently presented. The hp-adaptive scheme is obtained from the combination of a p-adaptive scheme with an anisotropic mesh adaptation method and applied to the resolution of the penalized Navier-Stokes equations. The robustness of the method is showed in practice with numerical experiments for different Mach regimes and Reynolds numbers in dimensions two and three.

3.1 Introduction

The class of immersed boundary methods is an efficient approach to deal with compressible viscous flows [50]. One drawback is the treatment of wall boundary conditions, which remains a complicated matter in general [51],[56],[67],[82]. In [8], the authors present the Immersed Boundary Method with Level Sets and Adapted Unstructured Meshes (IBM-LS-AUM) which is an immersed boundary method that combines the penalization method with an anisotropic adaptation of the mesh. The penalization method used in the reference consists in modifying the governing equations by the addition of penalization terms, resulting in the so-called Navier-Stokes Brinckman equations [21]. These penalization terms are handled as a source term in the numerical scheme and the solid is taken into account with a level set function. The h-adaptation method used in the reference is a metric-based method that adapts the mesh to the solid interface and to the physics of the solution. This method is able to correctly represent the boundary of the object which is of crucial importance when the wall boundary conditions are imposed by a penalization technique. The IBM-LS-AUM method therefore yields the benefits of an immersed boundary method, in particular the meshing process is simplified compared to body-fitted methods, and the anisotropic mesh adaptation combined with the penalization method simplifies the the treatment of wall boundary conditions and increases the accuracy of the representation of the surface and the solution around walls. It is applied as an iterative algorithm, where a first solution is computed, then the mesh is adapted with respect to the solution and to the level set, then the simulation is run again and the process is repeated until a qualitatively accurate solution is obtained.
In this chapter, the idea is to combine the IBM-LS-AUM with p-adaptation. Indeed, high order methods are now gaining more and more interest for realistic and large scale computational fluid dynamics simulations [88]. When high order methods can be implemented, they usually offer significant gains in term of computation time, due to the higher convergence rate [70]. In order to build high order methods that are efficient and robust, a general guideline is to use high order methods for problems where the solution is expected to be smooth, and low order methods where the solution is expected to be discontinuous. The optimal approach is then to use hp-adaptation, in order to use high order coarse elements where the solution is regular enough, and low order small elements where it is less regular. This approach leads to an optimal balance between the number of degrees of freedom and the level of accuracy of the simulation. The results presented in [16], show that it is now possible to construct a Residual Distribution scheme able to handle p-adaptation with elements (triangular or tetrahedral) of arbitrary orders. From these results, we propose here to implement mesh p-adaptation in the IBM-LS-AUM algorithm, thus resulting in an immerse boundary method combining hp-adaptation and

penalization within a Residual Distribution scheme.
The structure of the chapter is as follows. In section 3.2 we recall the principles of the penalization method for the compressible Navier-Stokes equations, in section 3.3 we recall the definitions and properties of a high order Residual Distribution scheme and present a penalized high order Residual Distribution scheme. In section 3.4 we first present the h-adaptation method used in our numerical scheme. In the second part of this section, we recall the results presented in [16]. The formulas proposed by the authors allow to implement efficiently p-adaptation within the frame of Residual Distribution schemes. In the third part of section 3.4, we present our new algorithm, as a high order element evolution of the IBM-LS-AUM algorithm. Finally, we present in section 3.5 some numerical results for subsonic, transonic and supersonic test cases at different Reynolds numbers in the two and three dimensions.

3.2 The penalization method for the compressible Navier-Stokes equations

We study in the present work the numerical solution of the compressible Navier-Stokes system of equations

$$\begin{aligned} &\nabla \cdot (\rho \mathsf{u}) = 0 \\ &\nabla \cdot (\rho \mathsf{u} \otimes \mathsf{u}) + \nabla p = \nabla \cdot \tau \\ &\nabla \cdot \big((\rho E + p)\mathsf{u}\big) = \nabla \cdot (\tau \cdot \mathsf{u} - \mathsf{q}) \end{aligned} \tag{3.1}$$

where ρ is the density, u is the fluid velocity, E is the specific total energy, p is the pressure, τ the viscous stress tensor and q the heat flux vector.

We suppose a calorically perfect gas, and the Navier-Stokes system is closed with the following two relations for energy

$$E = e + \frac{\mathsf{u} \cdot \mathsf{u}}{2} \quad \text{and} \quad e = \frac{1}{\gamma - 1} RT, \tag{3.2}$$

with the thermodynamic relation of state for a perfect gas

$$p = \rho R T \tag{3.3}$$

where $\gamma = 1.4$ is the specific heat ratio, T the gas temperature and R the gas constant which is $287 N.m/kg$ for sea-level air.

The viscous stress tensor is defined using the Stoke hypothesis

$$\tau = -\frac{2}{3}\mu(\nabla \cdot \mathsf{u})\mathbb{I} + \mu(\nabla^t \mathsf{u} + \nabla \mathsf{u}) \tag{3.4}$$

and the heat flux is computed using the Fourier law

$$\mathsf{q} = -\kappa \nabla T, \tag{3.5}$$

where κ is the thermal conductivity determined by the Prandtl number

$$\kappa = \frac{\gamma R \mu}{(\gamma - 1)\mathrm{Pr}}.$$

The system of conservation equations (3.1) can be expressed in vector form

$$\nabla \cdot \ \mathbf{f}(\mathbf{u}) = 0 \tag{3.6}$$

with the vector of unknowns $\mathbf{u}$ and the flux function $\mathbf{f}(\mathbf{u})$ defined by

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho \mathsf{u} \\ \rho E \end{pmatrix}, \qquad \mathbf{f}(\mathbf{u}) = \begin{pmatrix} \rho \mathsf{u} \\ \rho \mathsf{u} \otimes \mathsf{u} + p \ \mathrm{Id}_{d\times d} - \tau \\ (\rho E + p)\mathsf{u} - \tau \cdot \mathsf{u} + \mathsf{q} \end{pmatrix},$$

where $\mathbf{u} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^l$ is the vector of l conservative variables, Ω is an open set of $\mathbb{R}^d$, $d = 2, 3$, and $\mathbf{f} = (\mathbf{f}_1, ..., \mathbf{f}_d)$ is the vector of flux functions with $\mathbf{f}_i(\mathbf{u}) : \Omega \to \mathbb{R}^l$, $i = 1, ..., d$.

If the flux $\mathbf{f}(\mathbf{u})$ is split into the advective and viscous parts, the equation (3.6) can be written as

$$\nabla \cdot \ \mathbf{f}^a(\mathbf{u}) = \nabla \cdot \ \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) \tag{3.7}$$

with

$$\mathbf{f}^a(\mathbf{u}) = \begin{pmatrix} \rho \mathsf{u} \\ \rho \mathsf{u} \otimes \mathsf{u} + p \ \mathrm{Id}_{d\times d} \\ (E + p)\mathsf{u} \end{pmatrix} \quad \text{and} \quad \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{pmatrix} 0 \\ \tau \\ \tau \cdot \mathsf{u} - \mathsf{q} \end{pmatrix}.$$

The viscous flux function, being homogeneous with respect to the gradient of the conservative variables, can be written as

$$\mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u}) = \mathbb{K}(\mathbf{u}) \nabla \mathbf{u} \tag{3.8}$$

where the tensor $\mathbb{K} \in \mathbb{R}^{p\times d\times p\times d}$ is the viscous flux Jacobian defined by

$$\mathbb{K}(\mathbf{u}) = \frac{\partial \mathbf{f}^v(\mathbf{u}, \nabla \mathbf{u})}{\partial \nabla \mathbf{u}} \tag{3.9}$$

and for convenience, we denote $\mathbb{K}_{ij}$ the matrix

$$\mathbb{K}_{ij}(\mathbf{u}) = \frac{\partial \mathbf{f}_i^v(\mathbf{u}, \nabla \mathbf{u})}{\partial (\frac{\partial \mathbf{u}}{\partial x_j})}. \tag{3.10}$$

We use a penalization method to impose the no-slip boundary condition

$$\mathsf{u} = 0 \tag{3.11}$$

at the wall boundary of the solid. More precisely, the governing equations (3.1) are modified by the addition of penalization terms that enforce the no-slip boundary condition (3.11) on the solid. The solid is consequently represented by the penalization term and is considered as a porous media with very small intrinsic permeability. This method is called the penalization method [26] and the penalized equations (3.14) are called the Brinckman Navier-Stokes equations.

We present now the penalized Navier-Stokes system of equations. The derivation of this system of equations is detailed in [8].
We denote by $\frac{1}{\eta} >> 1$ the penalization parameter, by χ_S the characteristic function of the solid, by u_S the velocity imposed in the solid and by T_S the temperature imposed in the solid.
We define implicitly the signed distance function ϕ_S to the solid S by the relations

$$\begin{aligned} S &= \left\{x \in \mathbb{R}^d, \phi_S(x) < 0\right\}, \\ \partial S &= \left\{x \in \mathbb{R}^d, \phi_S(x) = 0\right\}, \\ {}^c\bar{S} &= \left\{x \in \mathbb{R}^d, \phi_S(x) > 0\right\}. \end{aligned} \tag{3.12}$$

The characteristic function is related to the distance function by the following equality, where H denotes the Heaviside function

$$\chi_S = H(-\phi_S). \tag{3.13}$$

We can now write the Navier-Stokes penalized system of equations

$$\begin{aligned} &\nabla \cdot (\rho \mathsf{u}) = 0, \\ &\nabla \cdot (\rho \mathsf{u} \otimes \mathsf{u}) + \nabla p = \nabla \cdot \tau + \frac{1}{\eta}\chi_S(\rho \mathsf{u} - \rho \mathsf{u}_S) \\ &\nabla \cdot \big((\rho E + p)\mathsf{u}\big) = \nabla \cdot (\tau \cdot \mathsf{u} - \mathsf{q}) + \frac{1}{\eta}\ \theta_S\ \chi_S\ \rho\ (e(T) - e(T_S)) + \frac{1}{\eta}\chi_S(\rho \mathsf{u} - \rho \mathsf{u}_S) \cdot \mathsf{u} \end{aligned} \tag{3.14}$$

where θ_S is a parameter set to 1 when the temperature T_S is enforced on the solid boundary, and set to 0 for a Neumann boundary condition on the temperature (adiabatic wall).

In vector form, theses equations can be written as

$$\nabla \cdot\ \mathbf{f}(\mathbf{u}) = S(\mathbf{u}) \tag{3.15}$$

where $S(\mathbf{u})$ is the source term containing the penalization terms of equation (3.14)

$$S(\mathbf{u}) = \frac{\mathbf{1}}{\eta}\chi_{\mathbf{S}} \begin{pmatrix} 0 \\ \rho \mathsf{u} - \rho \mathsf{u}_S \\ \theta_S\ \rho\ \big(e(T) - e(T_S)\big) + (\rho \mathsf{u} - \rho \mathsf{u}_S) \cdot \mathsf{u} \end{pmatrix}. \tag{3.16}$$

In the obstacle, we have $\chi_S = 1$ and so by identifying the terms of order $\frac{1}{\eta}$ we retrieve the boundary condition $\mathsf{u} - \mathsf{u}_S = 0$ and so, with $\mathsf{u}_S = 0$, we recover the boundary condition (3.11). Finally, with relations (3.2) we retrieve when $\theta_S = 1$ the boundary condition $T = T_S$ inside the wall boundary of the solid. With $\theta_S = 0$ we impose a no flux boundary condition.

On the boundary of the domain $\partial\Omega$, we impose the condition

$$\mathbf{f}(\mathbf{u}_h) = \mathbf{f}(\mathbf{u}^\partial) \tag{3.17}$$

where $\mathbf{f}(\mathbf{u}^\partial)$ is a Steger-Warming numerical flux and $\mathbf{u}_h$ is the numerical solution. The practical imposition of this condition in the numerical scheme is detailed in section 3.3.1.

3.3 Residual Distribution schemes for the penalized Navier-Stokes equations

3.3.1 General definitions

We are looking for an approximate solution to equations (3.14) with the boundary conditions (3.17). We suppose in the work presented here that Ω is a polyhedric open set, more precisely we suppose that $\bar{\Omega}$ is constituted of non degenerate triangles in dimension two and tetrahedrons in dimension three. We can then consider a finite decomposition of the domain

$$\bar{\Omega} = \underset{K\in\mathscr{T}_h}{\cup} K \tag{3.18}$$

where K is a simplex and $\mathscr{T}_h$ is a conformal triangulation of $\bar{\Omega}$. We associate to each simplex $K \in \mathscr{T}_h$ a finite element $(K, \mathbb{P}^k(K)^p, \Sigma_K)$ where Σ_K denotes the set of degrees of freedom of K and is defined by

$$\Sigma_K = \left\{ x \in \mathbb{R}^n, \lambda_j(x) \in \{0, \frac{1}{k}, ..., \frac{k-1}{k}, 1\}, 1 \leq j \leq n+1 \right\} \tag{3.19}$$

where $\lambda_j, j = 1, ..., d+1$ are the barycentric coordinates with respect to the vertices of K (the finite elements thus defined are affine equivalent to each other).

We can then define the finite element space

$$X_h = \left\{ \mathbf{u}_h \in C^1(\bar{\Omega})^p;\ \forall K \in \mathscr{T}_h,\ \mathbf{u}_{h|K} \in \mathbb{P}^k(K)^p, k \in \mathbb{N}^* \right\} \tag{3.20}$$

where we look for the discrete solution to equations (3.14)-(3.17).

We define now the basis of the vector space $\mathbb{P}^k(K)$. We will use in this chapter the Lagrange basis functions and the Bézier basis functions.
From the definition (3.19), any degree of freedom $a_\mu \in \Sigma_K$ can be written as

$$a_\mu = \frac{1}{k}\sum_{j=1}^{d+1} \mu_j a_j, \ \ \mu = (\mu_1, ..., \mu_{d+1}) \tag{3.21}$$

where $(a_j)_{j=1,...,d+1}$ are the vertices of K and $\frac{1}{k}(\mu_1, ..., \mu_{d+1})$ are the barycentric coordinates of a_μ with respect to the vertices of K.
The Lagrange basis function φ_μ associated to a_μ can then be written as

$$\varphi_\mu = \big(\prod_{j=1}^{d+1}(\mu_j!)\big)^{-1} \prod_{j=1,\ \mu_j\geq 1}^{d+1} \prod_{i=0}^{\mu_j - 1} \big(k\lambda_j - i\big) \tag{3.22}$$

and the Bézier basis function B_μ associated to a_μ can be written as

$$B_\mu = k!\big(\prod_{j=1}^{d+1}(\mu_j!)\big)^{-1} \prod_{j=1}^{d+1} \lambda_j^{\mu_j} \tag{3.23}$$

We remark that

$$\forall\ \mathbf{x} \in K, \ \sum_{a_\mu \in \Sigma_K} B_\mu(\mathbf{x}) = 1 \tag{3.24}$$

and that the functions B_μ are positive on K.

We denote by Σ_h the set of degrees of freedom of the finite elements

$$\Sigma_h = \underset{K\in\mathscr{T}_h}{\cup} \Sigma_K \tag{3.25}$$

and the approximate solution $\mathbf{u}_h$ writes

$$\mathbf{u}_h = \sum_{i\in\Sigma_h} \mathbf{u}_i \psi_i \tag{3.26}$$

where the functions ψ_i are the basis functions of X_h and the support of ψ_i is the set of elements K such that $i \in K$ and the restriction of ψ_i to K is a basis function (Lagrange or Bézier) of $\mathbb{P}^k(K)$.

We can now write the discrete equations of the residual distribution scheme. They are described by means of residuals that are defined for each degree of freedom of the elements $K \in \mathscr{T}_h$.
We first introduce the total residual of the element K

$$\Phi^K(\mathbf{u}_h) = \int_K \nabla \cdot \mathbf{f}(\mathbf{u}_h) dx, \tag{3.27}$$

computed in practice, thanks to the continuity of $\mathbf{u}_h \in X_h$, as

$$\Phi^K(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \, dx \tag{3.28}$$

where $\mathbf{n}$ is the exterior normal to ∂K.

We define as well the boundary total residual for a boundary element (edge in dimension two and face in dimension three) $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$

$$\Phi^\Gamma = \int_\Gamma \big(\mathbf{f}(\mathbf{u}^\Gamma) - \mathbf{f}(\mathbf{u}^h)\big) \cdot \mathbf{n} \, dx \ . \tag{3.29}$$

Once the total residual $\Phi^K(\mathbf{u}_h)$ is computed, it is distributed to the degrees of freedom $\sigma \in \Sigma_K$, which gives the nodal residuals that can be defined in generic form by

$$\Phi^K_\sigma = \beta^K_\sigma(\mathbf{u}_h)\Phi^K(\mathbf{u}_h) \tag{3.30}$$

and as well the total residual Φ^Γ is distributed to the degrees of freedom $\sigma \in \Sigma_\Gamma$

$$\Phi^\Gamma_\sigma = \beta^\Gamma_\sigma(\mathbf{u}_h)\Phi^\Gamma(\mathbf{u}_h). \tag{3.31}$$

The nodal residuals must satisfy the conservation constraints

$$\sum_{\sigma \in \Sigma_h} \Phi^K_\sigma = \Phi^K \tag{3.32}$$

and

$$\sum_{\sigma \in \Sigma_\Gamma} \Phi^\Gamma_\sigma = \Phi^\Gamma \ . \tag{3.33}$$

The crucial part of the method is the definition of the residuals Φ^K_σ according to the desired properties of the numerical scheme.
We detail the schemes used in this chapter for the Navier-Stokes equations in section 3.3.3.
For the boundary residuals, they are in practice defined as follows [2]

$$\Phi^\Gamma_\sigma = \int_\Gamma \psi_\sigma \big(\mathbf{f}(\mathbf{u}^\partial) - \mathbf{f}(\mathbf{u}_h)\big) \cdot \mathbf{n} \, dx, \tag{3.34}$$

where ψ_σ is the basis function of the element K such that $\Gamma \subset K$, associated to the degree of freedom σ (or equivalently the basis function of Γ associated to σ as they coincide on Γ) and where $\mathbf{f}(\mathbf{u}^\partial)$ is a numerical flux that depends on the boundary condition $\mathbf{u}^\partial$, the outward normal $\mathbf{n}$ and the local state $\mathbf{u}_h$. For the inflow/outflow Steger-Warming boundary condition (3.17), used in this chapter, the numerical flux is defined by

$$\mathbf{f}(\mathbf{u}^\partial) = \big(A(\mathbf{u}_h) \cdot \mathbf{n}\big)^+ \mathbf{u}^h + \big(A(\mathbf{u}_h) \cdot \mathbf{n}\big)^- \mathbf{u}_\infty \tag{3.35}$$

where $A = (A_1, ..., A_d)$ with

$$A_i(\mathbf{u}) = \frac{\partial \mathbf{f}_i(\mathbf{u})}{\partial \mathbf{u}} \tag{3.36}$$

and where $\mathbf{u}_\infty$ is the local state at infinity.

Finally the residual distribution scheme writes:

$$\forall \sigma \in \mathscr{T}_h, \sum_{K, \sigma \in \Sigma_K} \Phi_\sigma^K + \sum_{\Gamma, \sigma \in \Sigma_\Gamma} \Phi_\sigma^\Gamma = 0. \tag{3.37}$$

3.3.2 High order gradient reconstruction for Residual Distribution schemes

For several reasons described in [9], it is preferable to consider a numerical method that treats the advective and diffusive terms in the same single scheme. This means that a single solution is computed, and as the numerical solution is a piecewise polynomial in each element, the gradient of the numerical solution is in general not continuous across the interface of two adjacent elements. In the Residual Distribution framework, this is a problem, as the continuity of the normal component of the gradient of the solution is one of the main argument used to prove that the scheme is consistent with respect to the original equations [13]. So, one has to ensure the continuity of the gradient of the numerical solution. We can first cite two strategies based on the introduction of a numerical flux invoked in [9]. One approach could be to construct the numerical flux in a similar way as what has been done in [4] in the context of discontinuous Residual Distribution schemes for hyperbolic problems. However, as mentioned by the author of the reference, this would probably complexify the scheme and bring accuracy issues. The other approach could be to apply to Residual Distribution schemes the construction of the numerical flux proposed in [49] for Discontinuous Galerkin and Finite Volume schemes. However, this approach has not yet been tried for Residual Distribution schemes. We follow here another strategy, presented in [9], which proposes a high order and continuous reconstruction of the gradient for Residual Distribution schemes, while preserving the compactness of the scheme and such that the computation of the recovered high order gradient remains fast. The method is called "super convergent patch recovery" (SPR-ZZ) and was first introduced in [91].
We recall briefly the principle of the method. The complete description of the method can be found in [92] and its adaptation to Residual Distribution schemes can be found in [9].
The goal is to compute a unique value of $\nabla \mathbf{u}_h$ at the vertices of the mesh, such that $\nabla \mathbf{u}_h$ is of order $k+1$ like the solution $\mathbf{u}_h$, where k is the degree of the interpolating polynomial $\mathbf{u}_{h_{|K}}$ in the element K. The first step is to construct

the patch of the vertex (i.e. all the elements containing the vertex). Then $\nabla \mathbf{u}_h$ is written as a polynomial for each spatial component, and these polynomials are determined from the so-called super convergent sample points and by solving a minimization problem by a least-square type method. These super convergent points are specially positioned. Indeed, it is shown in [92], that for well chosen points, the gradient interpolated from its values at those special points has the same order of precision as the numerical solution. The result is proved for segments, quadrangles and hexahedrons (via tensor product), and numerical experiments in [9] prove that the reconstruction of the gradient is of high order for triangles and tetrahedrons too. The problem of finding the constitutive polynomials is well posed when enough super convergent points are available with respect to the number of degrees of freedom necessary to define the constitutive polynomials. In general, this is the case in the domain. In order to ensure that the reconstructed gradient is continuous across the interfaces, a unique value is defined at each high order degree of freedom (used to define the high order polynomial solution). The value of the reconstructed gradient at such degree of freedom is defined as the value of the polynomial constructed for the patch of the closest vertex. When the degree of freedom is shared by different patches, uniqueness is ensured by averaging the values obtained in the different patches. For the reconstruction of the gradient at a boundary node, if not enough super convergence points are available for the problem to be well-posed, the reconstructed gradient of the closest domain patch can be used to define the value of the reconstructed gradient at the boundary node. Finally, it is shown in [9] that the method is computationally cheap compared to the more classical (and lower order) Green-Gauss methods.

3.3.3 Computation of the nodal residuals for the Navier-Stokes equations

For clarity, we first present the computation of the nodal residuals in the case where the total residual does not include a source term.
As we have seen in section 3.3.2, we use a continuous approximation of the gradient of the numerical solution. The total residual is then expressed as

$$\Phi^K = \int_{\partial K} \left(\mathbf{f}^a(\mathbf{u}_h) - \mathbb{K}(\mathbf{u}_h) \widetilde{\nabla \mathbf{u}_h} \right) \cdot \mathbf{n} \, dx \tag{3.38}$$

where $\widetilde{\nabla \mathbf{u}_h}$ is the reconstructed gradient of section 3.3.2.

We present now the distribution schemes used in the present work to distribute this total residual to the degrees of freedom of K.

The Lax-Wendroff scheme

The Lax-Wendroff scheme is a simplified version of the SUPG scheme [16]. It is defined by

$$\Phi_\sigma^{K,LW} = \frac{\Phi^K}{N_{\text{dof}}^K} + \int_K \left(A\cdot\nabla\psi_\sigma\right) \mathcal{T} \left(A\cdot\nabla\mathbf{u}_h - \nabla\cdot(\mathbb{K}\widetilde{\nabla\mathbf{u}_h})\right) dx + \int_K \mathbb{K}\nabla\psi_\sigma\cdot\left(\nabla\mathbf{u_h} - \widetilde{\nabla\mathbf{u}_h}\right) \tag{3.39}$$

where N_{dof}^K is the cardinal of Σ_K, A is the Jacobian matrix of the advection flux defined in (3.35), ψ_σ is the basis function (Lagrange or Bézier, see section 3.3.1) associated to the degree of freedom σ and $\mathcal{T}$ is a scaling matrix defined by

$$\mathcal{T} = \frac{1}{N_{dim}}|K|\left(\sum_{i=1}^{N_{\text{dof}}^K} R_{\mathbf{n}_i}(\bar{\mathbf{u}}) \wedge_{\mathbf{n}_i}^+ (\bar{\mathbf{u}}) L_{\mathbf{n}_i}(\bar{\mathbf{u}}) + \sum_{j=1}^{N_{dim}} \mathbb{K}_{jj}(\bar{\mathbf{u}})\right)^{-1} \tag{3.40}$$

where N_{dim} is the spatial dimension of the domain, $|K|$ is the volume of the element K, $\bar{\mathbf{u}}$ is the average value of the vector of conservative variables on the element K, $R_{\mathbf{n}}$ and $L_{\mathbf{n}}$ are resp. the matrices of the right and left eigenvectors of the matrix $A\cdot\mathbf{n}$, $\wedge_{\mathbf{n}_i}^+$ is the diagonal matrix with $\max(\lambda_{\mathbf{n}}, 0)$ on the diagonal, where the $\lambda_{\mathbf{n}}$ are the eigenvalues of the matrix $A\cdot\mathbf{n}$, and the matrix $\mathbb{K}_{jj}$ is the matrix defined in (3.10). The vector $\mathbf{n}_i$ is defined by

$$\mathbf{n}_i = \frac{1}{N_{\text{dim}}} \int_{\partial K} \nabla\psi_i \, dx. \tag{3.41}$$

Finally, the last term of the right-hand side of (3.39) is an extra stabilization term for the diffusion, added to penalize the difference between the reconstructed continuous gradient $\widetilde{\nabla\mathbf{u}_h}$ and the discontinuous gradient $\nabla\mathbf{u}_h$ computed in the element K. The addition of this term improves the robustness of the iterative convergence of the scheme [9].

This scheme is linearity preserving but not monotonicity preserving. It will be used for the subsonic test case in section 3.5.

The limited stabilized Rusanov scheme

It is defined by:

$$\begin{aligned}\Phi_\sigma^{K,RV} = \widehat{\Phi}_\sigma^K + \epsilon_h^K(\mathbf{u}_h) \int_K \left(A\cdot\nabla\varphi_\sigma - \nabla\cdot(\mathbb{K}\nabla\varphi_\sigma)\right) \mathcal{T} \left(A\cdot\nabla\mathbf{u}_h - \nabla\cdot(\mathbb{K}\widetilde{\nabla\mathbf{u}_h})\right) dx \\ + \int_K \mathbb{K}\nabla\psi_\sigma\cdot\left(\nabla\mathbf{u_h} - \widetilde{\nabla\mathbf{u}_h}\right) dx \ .\end{aligned} \tag{3.42}$$

The hat symbol denotes the use of a limitation technique to increase the order of accuracy. Indeed, the scheme above is initially constructed from the first order monotone Rusanov scheme

$$\Phi_\sigma^K = \frac{\Phi^K}{N_{\text{dof}}^K} + \frac{1}{N_{\text{dof}}^K}\alpha_K \sum_{\sigma' \in \Sigma_K} (\mathbf{u}_\sigma - \mathbf{u}_{\sigma'}). \tag{3.43}$$

Then, a limitation procedure is applied to the distribution coefficients β_σ^K (3.30) so that they are uniformly bounded. It is shown in [10] that the limitation procedure can give a scheme with non physical oscillations and poor iterative convergence to the steady state solution. The following step is then to add a filtering term, which is the second term of (3.42), where the matrix $\mathcal{T}$ is the same as in (3.39). The term $\epsilon_h^K(\mathbf{u}_h)$ is a shock detector based on the variation of pressure, with $\epsilon_h^K(\mathbf{u}_h) \simeq 1$ in regions where the solution is smooth, and $\epsilon_h^K(\mathbf{u}_h) \simeq 0$ where the solution is discontinuous. We use the pressure detector of [16]

$$\epsilon_h^K(\mathbf{u}_h) = \max_{\sigma \in \Sigma_K} \left(\max_{K', \sigma \in \Sigma_{K'}} \frac{|\max_{\nu \in \Sigma_{K'}} p_\nu - \min_{\nu \in \Sigma_{K'}} p_\nu|}{|\max_{\nu \in \Sigma_{K'}} p_\nu| + |\min_{\nu \in \Sigma_{K'}} p_\nu| + \epsilon} \right) \tag{3.44}$$

where K and K' are elements of $\mathcal{T}_h$ and ϵ represents the machine epsilon (we take $\epsilon = 1e-16$).

As in the Lax-Wendroff scheme (3.39), the last term of the right-hand side of (3.42) is an extra stabilization term used to improve the iterative convergence.

The scheme (3.42) is both linearity and monotonicity preserving and it is the residual distribution scheme that we use for the transonic and supersonic test cases of section 3.5.

3.3.4 Practical computation of the total residual and the nodal residuals

In the particular case where the total residual includes the source term (3.16) that corresponds to the penalization terms, it is computed as follows

$$\begin{aligned} \Phi^K(\mathbf{u}_h) &= \int_K \left(\nabla \cdot \mathbf{f}(\mathbf{u}_h) - S(\mathbf{u}_h) \right) dx \\ &= \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \, dx - \int_K S(\mathbf{u}_h) \, dx \, . \end{aligned} \tag{3.45}$$

To compute the integral of the source term above, we need a numerical approximation χ_S^K of the indicator function χ_S for the element K. In practice,

we use the following definition:

$$\text{If } \max_{x \in V_K} \phi_S(x) < \varrho \text{ then } \chi_S^K = 1, \quad \text{otherwise } \chi_S^K = 0 \tag{3.46}$$

where V_K is the set of vertices of K and ϱ is a tolerance parameter (in practice we use $\varrho = 1e-10$).

As we have seen above, the computation of the total residual and the nodal residuals involves the computation of quadratures. From the results presented in [11] for the hyperbolic case that extend to the parabolic case, the quadrature formulas used for the computation of the total residual (3.45) need to be of sufficient order with respect to the order of the truncation error of the scheme, for both the boundary integral and the integral of the source term, while in the computation of the nodal residuals the quadrature formulas used for the computation of the stabilization terms in (3.39) and (3.42) can be less accurate.

3.4 hp-adaptation

We recall here the anisotropic mesh adaptation method for the penalized Navier-Stokes equations presented in [8] and the p-adaptive Residual Distribution scheme presented in [16] for hyperbolic problems. The general idea is to combine these two methods to construct a hp-adaptive Residual Distribution scheme for the penalized Navier-Stokes equations which is applied to the test cases of section 3.5. We first recall the principle of metric based mesh adaptation then we present the p-adaptive Residual Distribution scheme.

3.4.1 Anisotropic h-adaptation

We detail in the following the algorithm used for the anisotropic mesh adaptation. The general idea is to adapt the mesh to both the flow features and to the 0 level set of the function ϕ_S (3.12).

Construction of the metric tensor for mesh adaptation to the flow features

We recall the following theorem proved in [33] which is at the basis of the anisotropic mesh adaptation method used in this work.

Théorème 3.1. *We denote by $\pi_h u$ the $\mathbb{P}^1$ interpolant of a function $u \in C^2(\Omega)$ and $K \in \mathscr{T}_h$ denotes a simplex. The interpolation error on K can be controlled by the following bound*

$$\|u - \pi_h u\|_{\infty,K} \leqslant c \max_{x \in K} \max_{e \in E_K} (e, |H_u(x)|e) \tag{3.47}$$

where E_K is the set of edges of the element K, the constant c is related to the spatial dimension d, $H_u(x)$ defines the Hessian matrix of u at the point x, $(.,.)$ denotes the usual scalar product in $\mathbb{R}^d$ and where we define the matrix $|H_u(x)|$ as follows: as $H_u(x)$ is a symmetric matrix, we can decompose it in an orthonormal basis as (with $d = 3$)

$$H_u(x) = T \begin{pmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{pmatrix} T^t \tag{3.48}$$

and we define the matrix $|H_u(x)|$ by

$$|H_u(x)| = T \begin{pmatrix} |\mu_1| & 0 & 0 \\ 0 & |\mu_2| & 0 \\ 0 & 0 & |\mu_3| \end{pmatrix} T^t. \tag{3.49}$$

This bound is impractical to compute, so in practice [47] we construct from the matrix $|H_u|$ a metric tensor $\bar{M}(K)$ that verifies

$$\max_{x \in K}(e, |H_u(x)|e) \leqslant (e, \bar{M}(K)e), \quad \forall e \in E_K. \tag{3.50}$$

This gives the interpolation error bound

$$\|u - \pi_h u\|_{\infty,K} \leqslant c \max_{e \in E_K}(e, \bar{M}(K)e) \tag{3.51}$$

and thus we define the interpolation error ϵ_K on the element K by the following formula

$$\epsilon_K := c \max_{e \in E_K}(e, \bar{M}(K)e) \ . \tag{3.52}$$

If we define for all elements a global maximum level ϵ of the error of interpolation on the mesh, then the edges of K must verify

$$c(e, \bar{M}(K)e) = \epsilon, \forall e \in K. \tag{3.53}$$

Consequently, with the metric tensor $M(K)$ defined by

$$M(K) = \frac{c}{\epsilon}\bar{M}(K), \tag{3.54}$$

the edges of K must verify

$$(e, M(K)e) = 1, \forall e \in K. \tag{3.55}$$

As the distance $l_{M(K)}$ with respect to the metric tensor $M(K)$ is defined by

$$l_{M(K)}(x) = \sqrt{x^t M x}, \tag{3.56}$$

the relation (3.55) means that the edges of the mesh must verify

$$\left(l_{M(K)}(e)\right)^2 = 1. \tag{3.57}$$

In other words, the optimal mesh will be the unit mesh with respect to the metric $M(K)$.

In practice, the metric M is not defined on the elements but is defined at the vertices of the mesh [47]. The values of the metric at the other degrees of freedom are computed by interpolation from the values at the vertices [48].

Metric definition for mesh adaptation to the flow

We define now the metric used to adapt the mesh to the flow features in dimension three (in the work presented here, u is the x-component of velocity, but it could be, for example, the density or the pressure).
For this purpose, we first consider the approximation $\widetilde{|H(u)|}$ of the absolute value of the Hessian matrix $|H(u)|$ decomposed in diagonal form in an orthonormal basis as

$$\widetilde{|H(u)|} = \mathcal{R} \begin{pmatrix} |\lambda_1| & 0 & 0 \\ 0 & |\lambda_2| & 0 \\ 0 & 0 & |\lambda_3| \end{pmatrix} \mathcal{R}^t \tag{3.58}$$

and from there, we define the metric M_f for the adaptation to the flow features [47] by

$$M_f = \mathcal{R} \begin{pmatrix} \widetilde{\lambda_1} & 0 & 0 \\ 0 & \widetilde{\lambda_2} & 0 \\ 0 & 0 & \widetilde{\lambda_3} \end{pmatrix} \mathcal{R}^t, \tag{3.59}$$

where $\widetilde{\lambda_i} = \min\left(\max\left(\frac{c|\lambda_i|}{\epsilon}, \frac{1}{h_{max}^2}\right), \frac{1}{h_{min}^2}\right)$, with h_{min} and h_{max} respectively the minimal and maximal size of the edges of the elements of the mesh, c the constant of relation (3.47) and ϵ is the global error bound of relation (3.53).
As described in [47], the introduction of the coefficients h_{min} and h_{max} in the definition of the metric allows to impose edge lengths that are neither too big nor too small.

Metric definition for mesh adaptation to the level-set

In [32], the authors present a method to compute the signed distance function of the vertices of a computational mesh to a polyhedral domain with only the knowledge of its boundary supplied as a mesh (edges in dimension two and triangles in dimension three) supposed orientable. The computation of the signed distance function is based on an approximate solution to the unsteady eikonal equation.

In the work presented here, we suppose that with this method we have computed for each vertex x of the mesh $\mathscr{T}_h$ the distance $d(x, \partial S)$ of x to the boundary of S (∂S is the zero level set of ϕ_S as defined in section 3.2). We can then consider $\pi_h\phi_S$ the piecewise affine interpolant of the function ϕ_S and like for ϕ_S, we define the zero level set of the function $\pi_h\phi_S$ and we denote it by ∂S_h.
The goal of the mesh adaptation method presented in this section is to adapt the mesh $\mathscr{T}_h$ so that the zero level set of $\pi_h\phi_S$ obtained by $\mathbb{P}^1$ interpolation is as close as possible to the zero level set of the function ϕ_S in terms of the Hausdorff distance and at the same time the mesh needs to provide an accurate representation of the boundary ∂S. In order to achieve this, we follow the work presented in [40] and we use the following metric for the adaptation of the mesh to the 0 level-set of ϕ_S

$$M_{ls} = R \begin{pmatrix} \frac{1}{\epsilon^2} & 0 & 0 \\ 0 & \frac{|\lambda_1|}{\epsilon} & 0 \\ 0 & 0 & \frac{|\lambda_2|}{\epsilon} \end{pmatrix} R^t, \tag{3.60}$$

with $R = (\nabla\pi_h\phi_S, v_1, v_2)$ where (v_1, v_2) is a basis of the tangent plane to ∂S_h, where λ_1 and λ_2 are the eigenvalues of the Hessian matrix of the level-set $\pi_h\phi_S$ and where ϵ is a tolerance parameter. With this metric, we impose edge sizes that are small in the direction normal to ∂S and that are proportional to the curvature of ∂S which provides an accurate representation of ∂S. As we want to be accurate in the vicinity of ∂S, the metric above is imposed for the vertices that are close to ∂S and for the other vertices, we use a metric that increases linearly the sizes of the edges from $h_{\min}$ to $h_{\max}$ with respect to the distance from ∂S.

Mesh adaptation to the flow features and to the level-set with metric intersection

In the case where two metrics $\mathscr{M}_1(x)$ and $\mathscr{M}_2(x)$ are defined at the same point x of the mesh, in order to adapt the mesh with respect to these two metrics we resort to the procedure of metric intersection [47]. The idea is to find a common basis in which the two metrics $\mathscr{M}_1(x)$ and $\mathscr{M}_2(x)$ are simultaneously diagonal, which is possible as $\mathscr{M}_1(x)$ and $\mathscr{M}_2(x)$ are positive-definite matrices. Thus, if we denote P the common change of basis matrix that allows for the simultaneous reduction of $\mathscr{M}_1(x)$ and $\mathscr{M}_2(x)$, we can write the matrices $\mathscr{M}_1(x)$

and $\mathscr{M}_2(x)$ as (in the three dimensional case)

$$\begin{aligned}\mathscr{M}_1(x) &= P(x)\begin{pmatrix}\lambda_1(x) & 0 & 0\\ 0 & \lambda_2(x) & 0\\ 0 & 0 & \lambda_3(x)\end{pmatrix}P^t(x),\\ \mathscr{M}_2(x) &= P(x)\begin{pmatrix}\mu_1(x) & 0 & 0\\ 0 & \mu_2(x) & 0\\ 0 & 0 & \mu_3(x)\end{pmatrix}P^t(x)\end{aligned} \tag{3.61}$$

where $\lambda_1, \lambda_2, \lambda_3,\ \mu_1, \mu_2, \mu_3 > 0$.
The intersection of the two metrics is defined by

$$\mathscr{M}_1 \cap \mathscr{M}_2(x) = P(x)\begin{pmatrix}sup(\lambda_1(x), \mu_1(x)) & 0 & 0\\ 0 & sup(\lambda_2(x), \mu_2(x)) & \vdots\\ 0 & 0 & sup(\lambda_3(x), \mu_3(x))\end{pmatrix}P^t(x) \tag{3.62}$$

which defines a metric tensor as it is symmetric positive-definite.
In the work presented here, we intersect the metric M_f for the adaptation to the flow features computed in section 3.4.1 with the metric M_{ls} for the adaptation to the level-set computed in section 3.4.1. The resulting mesh is consequently adapted to both the flow features and to the level-set.

3.4.2 p-adaptation for Residual distribution scheme

We recall here the principles of p-adaptation for continuous Residual Distribution schemes presented in [16]. In the above reference, the authors present formulas that allow to split elements of arbitrary high order into affine elements within a Residual Distribution scheme. The idea in this chapter will be to apply the method by splitting $\mathbb{P}^2$ elements into $\mathbb{P}^1$ elements in the penalized zone and when relevant in zones with strong variations of pressure.

A Lax-Wendroff like theorem

We recall the Lax-Wendroff theorem [13] and its implications for p-adaptation in Residual Distribution schemes [16].

Théorème 3.2. *Assume the family of meshes $\mathscr{T} = (\mathcal{T}_h)_h$ is regular. For K an element or a boundary element of $\mathcal{T}_h$, we assume that the residuals $\{\Phi_\sigma^K\}_{\sigma\in K}$ satisfy:*

- *For any $M \in \mathbb{R}^+$, there exists a constant C which depends only on the family of meshes $\mathcal{T}_h$ and M such that for any $\mathbf{u}_h \in Z_h$ with $||\mathbf{u}_h||_\infty \leq M$, then*

$$\|\Phi_\sigma^K(\mathbf{u}_{h|K})\| \leq C \sum_{\sigma,\sigma'\in\Sigma_K} |\mathbf{u}_\sigma - \mathbf{u}_{\sigma'}| \tag{3.63}$$

- *They satisfy the conservation property* (3.32)-(3.33).

Then if there exists a constant C_{max} *such that the solutions of the scheme* (3.37) *satisfy* $||\mathbf{u}_h||_\infty \leq C_{max}$ *and a function* $\mathbf{V} \in L^2(\Omega)^p$ *such that* $(\mathbf{u}_h)_h$ *(or at least a sub-sequence) verifies* $lim_h||\mathbf{u}_h - \mathbf{v}||_{L^2_{loc}(\Omega)^p} = 0$, *then* $\mathbf{v}$ *is a weak solution of* (3.6).

As shown in [16], the two important arguments for the work presented here are the following.

The first argument is that the conservation relations (3.32)-(3.33) are required at the element level.

The second argument is that in the proof of the theorem, what matters is to have the following relation on any edge:
If $\Gamma = \Gamma'$ *is the same face shared by respectively the two adjacent elements* K *and* K', *then we have* $\mathbf{n}_{|\Gamma} = -\mathbf{n}_{\Gamma'}$ *and consequently*

$$\int_\Gamma \varphi_\sigma \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} \quad + \int_{\Gamma'} \varphi_\sigma \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} = 0, \tag{3.64}$$

where $\varphi_\sigma \in \mathbb{P}^k(K)$ *is the basis function associated to the node* σ.

The quantities (3.27)-(3.29) are computed with quadrature formulas:
For any element $K \in \mathscr{T}_h$,

$$\sum_{\sigma\in\Sigma_K} \Phi^K_\sigma(\mathbf{u}_h) = \int_{\partial K} \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n} \tag{3.65a}$$

and any boundary element $\Gamma \in \mathscr{T}_h \cap \partial\bar{\Omega}$,

$$\sum_{\sigma\in\Sigma_\Gamma} \Phi^\Gamma_\sigma = \int_\Gamma \left(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}\left(\mathbf{u}_h\right)\cdot\mathbf{n}\right) dx \tag{3.65b}$$

where here $\int$ signifies that the integral is computed with a quadrature formula, which reads

1. On elements:

$$\int_{\partial K} \mathbf{f}(\mathbf{u}^h)\cdot\mathbf{n} = \sum_{e \text{ edge/face } \subset \partial K} \left(|e| \sum_q \omega_q \mathbf{f}(\mathbf{u}_q)\cdot\mathbf{n}\right)$$

2. On boundary elements:

$$\int_\Gamma \left(\mathcal{F}(\mathbf{u}_h, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_h)\cdot\mathbf{n}\right) dx = |\Gamma| \sum_q \omega_q \left(\mathcal{F}(\mathbf{u}_q, \mathbf{u}_-, \mathbf{n}) - \mathbf{f}(\mathbf{u}_q)\cdot\mathbf{n}\right)$$

where ω_q and $\mathbf{u}_q$ are respectively the weight and the value of the solution at the quadrature point q.
So, the easiest way to implement the relation (3.64) is to use quadrature formulas such that the quadrature points coincide at the interface Γ.

The total residual $\Phi^K = \int_{\partial K} \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n}$ is computed with a quadrature formula as we have seen. In [16], the authors present formulas that express this total residual as a weighted sum of total residuals computed on the sub-elements of K. These general formulas, with the two arguments exposed above, make it possible to construct a p-adaptive Residual Distribution scheme with polynomials of arbitrary orders in dimension two and three. We recall now these formulas in the specific case of a quadratic interpolation.

Subdivision formula in 2d

We consider a triangle K and a quadratic interpolant $\mathbf{f}^{(2)}$ of the flux $\mathbf{f}$ defined on K. The vertices of K are denoted by 1,2,3 and the mid-points of the edges are denoted by 4,5,6. We subdivide K into the sub-triangles $K_1 = (1,4,6)$, $K_2 = (4,2,5)$, $K_3 = (5,3,6)$ and $K_4 = (4,5,6)$, as shown in figure 3.1. If we denote by $\lambda_i, i = 1,2,3$ the barycentric coordinates corresponding respectively to the vertices $i = 1,2,3$ then the linear interpolant of the flux $\mathbf{f}$ is defined by

$$\mathbf{f}^{(1)} = \sum_{i=1}^{3} \mathbf{f}(\mathbf{u}_i)\lambda_i$$

and its quadratic interpolant is defined by

$$\mathbf{f}^{(2)} = \sum_{i=1}^{6} \mathbf{f}(\mathbf{u}_i)\varphi_i$$

where:

$$\begin{array}{lll} \varphi_i = (2\lambda_i - 1)\lambda_i, & \text{for } i = 1,2,3 & \\ \varphi_4 = 4\lambda_2\lambda_1, & \varphi_5 = 4\lambda_3\lambda_2, & \varphi_6 = 4\lambda_1\lambda_3\,. \end{array}$$

We have the following formula for the quadratic interpolant in the two dimensional case

$$\begin{aligned} \int_K \text{div } \mathbf{f}^{(2)} dx = \frac{2}{3}\Big(\int_{K_1} \text{div } \mathbf{f}^{(1)} dx + \int_{K_2} \text{div } \mathbf{f}^{(1)} dx \\ + \int_{K_3} \text{div } \mathbf{f}^{(1)} dx \Big) + 2 \int_{K_4} \text{div } \mathbf{f}^{(1)} dx \end{aligned} \tag{3.66}$$

where $\mathbf{f}^{(1)}$ denotes the $\mathbb{P}^1$ interpolant of the flux in each of the sub-triangles of figure 3.1.

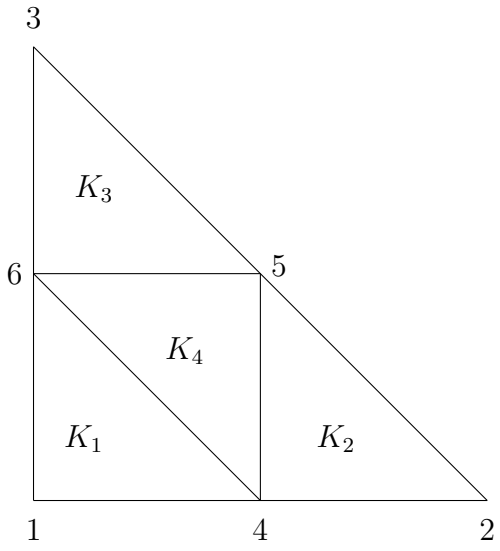


Figure 3.1: Subdivided triangle K.

Subdivision formula in 3d

We consider again a quadratic interpolant $\mathbf{f}^{(2)}$ of the flux $\mathbf{f}$, but now on the tetrahedron (figure 3.2). The formula in the 2d case can not be extended to the 3d case, due to a property of the $\mathbb{P}2$ Lagrange basis functions on the tetrahedron in 3d:

$$\int_K \nabla\varphi_j = 0, \;\; j = 4, 5, 6. \tag{3.67}$$

The consequence is that the degrees of freedom $4, 5, 6$ will not contribute to the sum of the nodal residuals, and so the numerical solution will often blow up. The solution is to replace the Lagrange basis by the Bézier basis (3.23), as the integrals $\int_K \nabla B_\sigma, \sigma = 1, ..., 6$ are non zero. As said previously, $\sum_{\sigma\in\Sigma_K} B_\sigma = 1$ and $B_{|K} \geq 0$. Other properties of the Bézier basis can be found in [15].
With $\mathbf{f}^{(2)}$ the Bézier approximation of the flux $\mathbf{f}$, we have the following formula in the three dimensional case

$$\begin{aligned}\int_K \text{div } \mathbf{f}^{(2)} dx = \frac{1}{2}\Big(&\int_{K_1} \text{div } \tilde{\mathbf{f}}^{(1)} dx + \int_{K_2} \text{div } \tilde{\mathbf{f}}^{(1)} dx \\ &+ \int_{K_3} \text{div } \tilde{\mathbf{f}}^{(1)} dx + \int_{K_4} \text{div } \tilde{\mathbf{f}}^{(1)} dx \Big)\end{aligned} \tag{3.68}$$

where

- on K_1, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_1\lambda_1 + \mathbf{f}_5\lambda_2 + \mathbf{f}_{10}\lambda_4 + \mathbf{f}_7\lambda_3$
- on K_2, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_2\lambda_2 + \mathbf{f}_6\lambda_3 + \mathbf{f}_5\lambda_1 + \mathbf{f}_8\lambda_4$

- on K_3, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_3\lambda_3 + \mathbf{f}_9\lambda_4 + \mathbf{f}_6\lambda_2 + \mathbf{f}_7\lambda_1$
- on K_4, $\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_8\lambda_2 + \mathbf{f}_9\lambda_3 + \mathbf{f}_{10}\lambda_1 + \mathbf{f}_4\lambda_4$

and where the sub-tetrahedrons K_1, K_2, K_3 and K_4 are obtained by subdivision of the tetrahedron K and are defined by

- K_1=(1,5,7,10)
- K_2=(5,2,6,8)
- K_3=(7,6,3,9)
- K_4=(10,8,9,4)

as shown in figure 3.2.

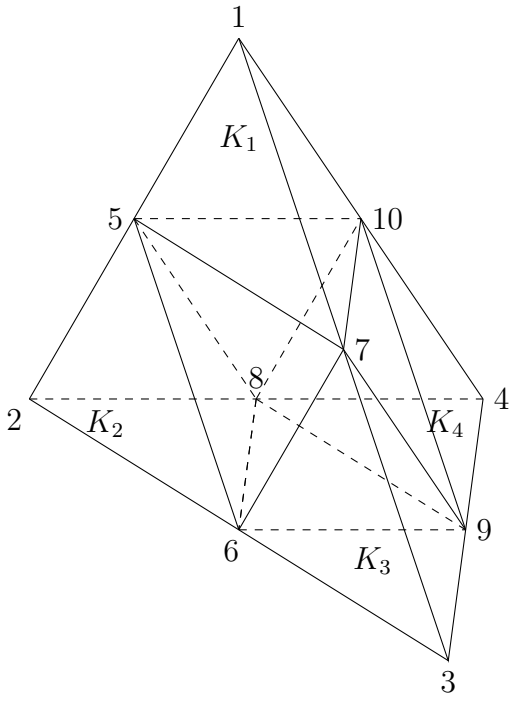


Figure 3.2: Subdivided tetrahedron K.

Nodal residuals for p-adaptation

We denote here S the set of $\mathbb{P}^1$ sub-elements of a subdivided element $K \in \mathscr{T}_h$ as described above and we denote $\gamma_\xi, \xi \in S$ the coefficients given by the formula (3.66) or (3.68) according to the dimension. As shown in [16], the residuals

$\Phi_\sigma^K, \sigma \in \Sigma_K$ are modified so that the conservation relations (3.32)-(3.33) are verified (see [16] for the proof):

$$\Phi_\sigma^K := \sum_{\xi \in S, \sigma \in \Sigma_\xi} \alpha_\xi \Phi_\sigma^\xi. \tag{3.69}$$

Therefore, we define (and compute) the contribution of a node σ in the element K computed in the sub-element ξ as

$$\Phi_\sigma^{K,\xi} = \alpha_\xi \ \Phi_\sigma^\xi \tag{3.70}$$

and similarly we define the contribution of a node σ in the boundary element Γ computed in the sub-face ς as

$$\Phi_\sigma^{\Gamma,\varsigma} \ = \alpha_\xi \ \Phi_\sigma^\varsigma \, . \tag{3.71}$$

Finally, in the case of an implicit scheme, the Jacobian of the residuals is modified and we have, respectively in the domain and on the boundary (with the same notation as above)

$$\begin{aligned} \frac{\partial \Phi_\sigma^{K,\xi}(\mathbf{u}_h)}{\partial \mathbf{u}_j} &= \alpha_\xi \ \frac{\partial \Phi_\sigma^\xi(\mathbf{u}_h)}{\partial \mathbf{u}_j}, \\ \frac{\partial \Phi_\sigma^{\Gamma,\varsigma}(\mathbf{u}_h)}{\partial \mathbf{u}_j} &= \alpha_\xi \ \frac{\partial \Phi_\sigma^\varsigma(\mathbf{u}_h)}{\partial \mathbf{u}_j}. \end{aligned} \tag{3.72}$$

3.5 Numerical test cases

We present now some tests cases that use simultaneously the techniques presented above, that is we combine anisotropic mesh adaptation with a p-adaptive Residual Distribution scheme to solve the penalized Navier-Stokes equations. We use the following protocol:

1. In the solid zone S, the term χ_S (3.13) is positive and we call this zone the penalized zone (more precisely the penalized zone is defined with the approximation (3.46) of χ_S, but to simplify the presentation, we consider here χ_S).
   This zone is delimited by the 0 level-set of ϕ_S, denoted by ∂S (see relation 3.12).
   The mesh is initially refined only around ∂S to represent the interface of the object around which the flow is studied.

2. The solution is first computed with this initial mesh, then the mesh is adapted with respect to the solution obtained (the Mach number) and to the 0 level-set, as described in section 3.4.1.

3. Then a new solution is computed on the adapted mesh and the mesh is again adapted (to the new solution and to the level-set).
   For the test cases presented here, three adaptation cycles have been performed and the mesh obtained after three adaptations is called the final mesh.

4. On the final mesh, we run one final computation.
   This computation is started with $\mathbb{P}^1$ elements everywhere, then after a small number of iterations that depends on the test case, we use the shock detector (3.44) to detect the zones where there is a strong variation of pressure. The elements in these zones are $\mathbb{P}^1$ elements.
   The penalized zone is composed of $\mathbb{P}^1$ elements too.
   All the other elements (outside the shock zones and outside the penalized zone) are $\mathbb{P}^2$ elements.

We follow the protocol described above for all test cases except for the NACA test cases where in step 4 *we do not use the shock detector as we use* $\mathbb{P}^2$ *elements everywhere except in the penalized zone as shown in figure 3.4.*

The essential idea we want to convey here is that these test cases confirm experimentally that hp-adaptation can be used in the frame of Residual Distribution schemes. Therefore we voluntary do not go into a comparison of penalized versus body-fitted mesh and high-order versus low order elements, as this is not our goal here (the reader can find the justifications and advantages of these approaches in [8] and [16]).

3.5.1 Naca0012 test cases in two dimension

We study the steady flow over a naca0012 airfoil at speeds varying from subsonic to hypersonic regimes. The domain is a disc of diameter 40 initially adapted only to the level-set zero of the Naca wing and constituted of about 45000 vertices (see figure 3.3). For all speed regimes (subsonic, transonic, supersonic and hypersonic), the simulation is run with this initial mesh, then three cycles of h-adaptation of the mesh are made, through which the mesh is adapted to both the physical solution and to the level-set zero of the Naca wing.
For these test cases, we use $\mathbb{P}^2$ Lagrange elements (3.22) everywhere, except in the penalized zone where $\mathbb{P}^1$ Lagrange elements are used.
Indeed, in order to demonstrate the robustness of the method, we want to show that we can use high order elements with the penalized Navier-Stokes equations within a Residual Distribution scheme at low to very high speed regimes and so we do not use a shock detector.

We detail now the boundary conditions, initial solutions and the characteristics of the final mesh for respectively the subsonic, transonic and supersonic test cases. We use the Steger-Warming boundary conditions (see section 3.3.1) for the boundary of the domain, with the following values of (ρ, u, v, p) for the different speed regimes:

- Subsonic: $(\rho, u, v, p) = (1, 0.5, 0, 0.7143)$,
- Transonic: $(\rho, u, v, p) = (1, 0.85, 0, 0.7143)$,
- Supersonic: $(\rho, u, v, p) = (1, 1.5, 0, 0.7143)$,
- Hypersonic: $(\rho, u, v, p) = (1, 5.0, 0, 0.7143)$.

For the five speed regimes, the initial solutions are set at the same values as the respective boundary conditions. Finally, for all speed regimes the Reynolds number is set at 5000.

The parameters for the mesh adaptation (see section 3.4.1) during the three adaptation cycles are the following, for both physical and level-set adaptation:

- $\epsilon = 0.001$,
- $h_{\min} = 0.001$,
- $h_{\max} = 0.5$.

The final meshes contain around 45000 vertices. The results are shown in figures 3.5 and 3.6, where the final mesh and the Mach number are presented for each speed regime. We can conclude that the results we obtain are qualitatively satisfactory. Indeed, the positions of the shocks and their evolutions in the transonic, supersonic and hypersonic test cases seem physically admissible.

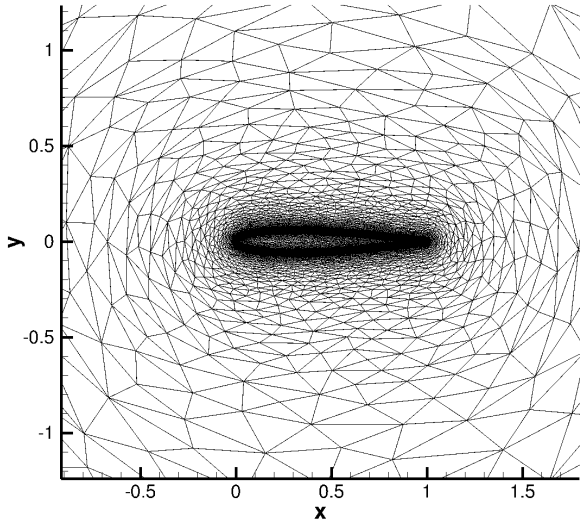


Figure 3.3: Zoom on the initial mesh for test cases 3.5.1. The mesh is only refined around the level-set zero of the Naca wing.

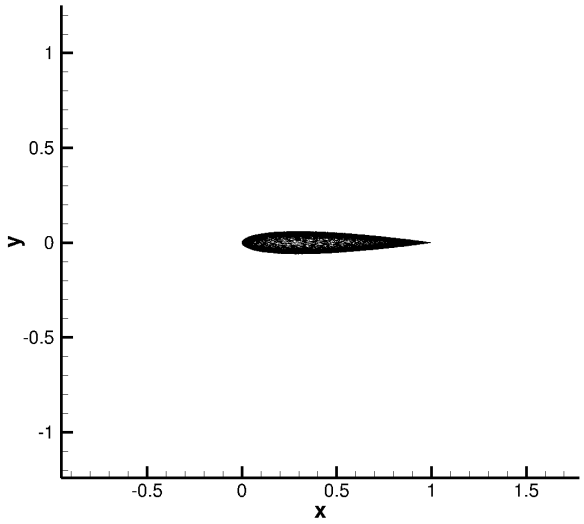


Figure 3.4: Penalized zone (in dark), contains $\mathbb{P}^1$ elements. The rest of the mesh contains $\mathbb{P}^2$ elements.

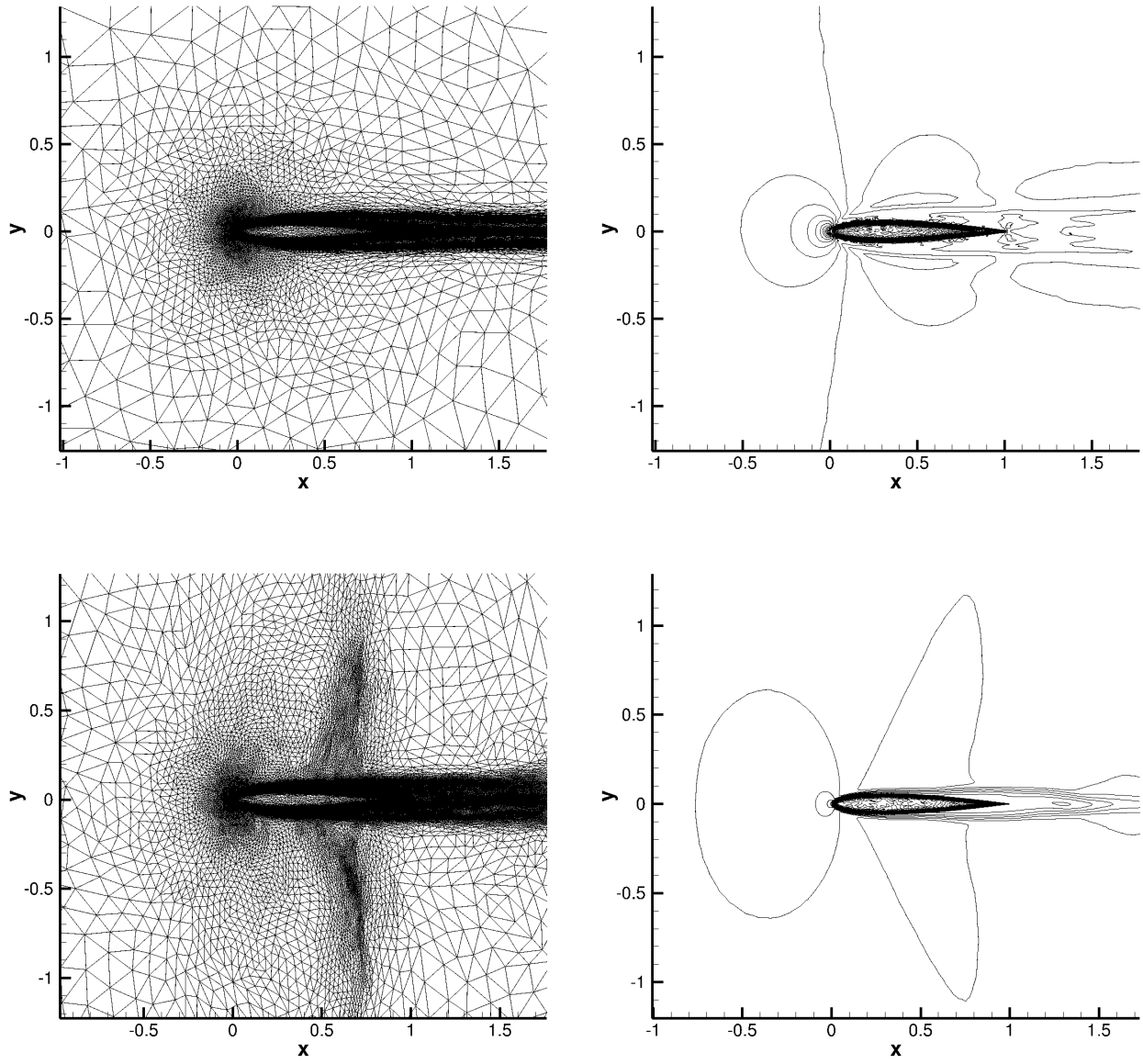


Figure 3.5: Final adapted meshes and solutions (Mach numbers) for test case 3.5.1 (subsonic and transonic regimes).

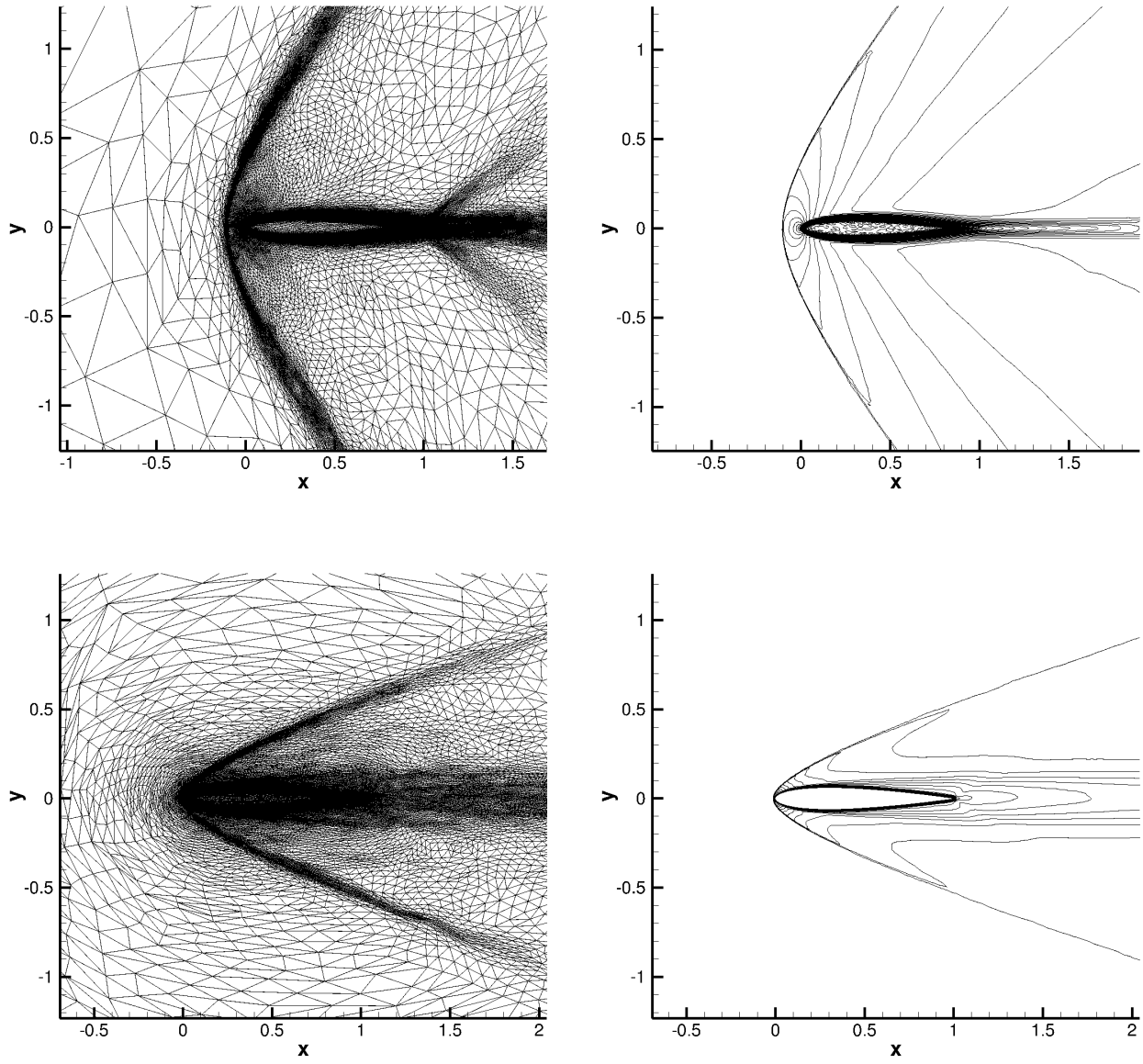


Figure 3.6: Final adapted meshes and solutions (Mach numbers) for test case 3.5.1 (supersonic and hypersonic regimes).

3.5.2 Two dimensionnal supersonic triangle test case

We study now the steady flow field over a triangle moving at a supersonic speed. The triangle height is h=0.5, its half angle is θ=20 degrees and its left apex is located at $(0.5, 1)$. The triangle is centered in a disc of diameter 40. At the boundary of the disc we impose the Steger-Warming boundary conditions with $(\rho, u, v, p) = (1, 2.366, 0, 1)$ and the variables are initialized with the values $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 2.366, 0, 1)$. In the triangle, the non dimensional temperature T_S (right hand-side of third equation in 3.14) is set at 3. The Reynolds number is set at $Re_\infty = 50000$. We choose these specific values in order to compare our solution with the results obtained in [26], as we will see below. The initial mesh is made of about 40000 vertices. We use the same parameters for the mesh adaptation as for the NACA test cases of section 3.5.1:

- $\epsilon = 0.001$,
- $h_{\min} = 0.001$,
- $h_{\max} = 0.5$.

On the final mesh, constituted of approximately 40000 vertices like the initial mesh, we use the shock detector (3.44) to detect the zones where the pressure presents strong variations. The final mesh is then constituted of Lagrange $\mathbb{P}^2$ elements (3.22) everywhere except in the shock and penalized zone which contain $\mathbb{P}^1$ elements. The contour of the $\mathbb{P}^1$ element zone is shown in figure 3.7.

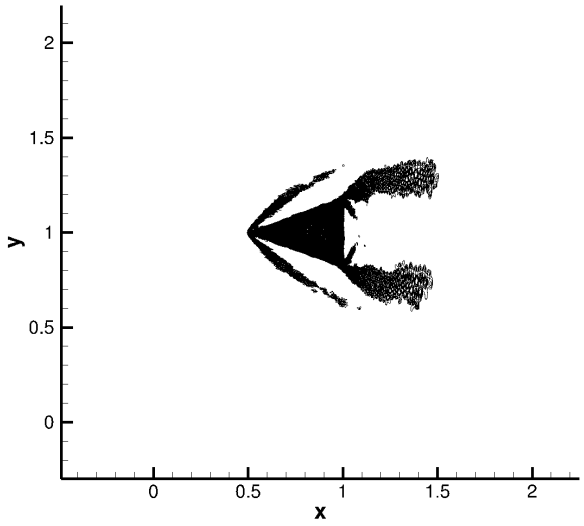


Figure 3.7: The dark zone (penalized zone + zone with strong shock) contains $\mathbb{P}^1$ elements, the rest of the mesh contains $\mathbb{P}^2$ elements (test case 3.5.2).

The mesh and solution (x-component of velocity) are displayed in figure 3.8. *The shock detector* (3.44) *is based on the variation of pressure, whereas the anisotropic adaptation is based on the variation of the x-component of velocity. As a consequence, the zones refined by the mesh adaptation (left frame of figure 3.8) do not exactly coincide with the zones detected by the shock detector (figure 3.7). To make them coincide, we could of course have chosen to use the same variable. But we think that choosing different variables demonstrate the flexibility and the robustness of the method.*

Now, following the study proposed in [26], we compute the pressure at a specific position (line $y = 1.44$) and show that our approach is in agreement with the results presented in this reference. A comparison between the results obtained in [26] and the results obtained with the penalized Residual Distribution scheme with $\mathbb{P}^1$ elements was presented in [8] and it was shown that the results obtained in the latest reference were in good agreement with the results obtained in [26]. From these results, we propose now a comparison between the pressure obtained on line $y = 1.44$ with $\mathbb{P}^1$ elements, hence serving as a reference, and the results obtained here with our penalized p-adaptive Residual Distribution scheme with the configuration of elements described above. We observe that the results obtained with p-adaptation are in good agreement with the result obtained with $\mathbb{P}^1$ elements, as shown in figure 3.9.

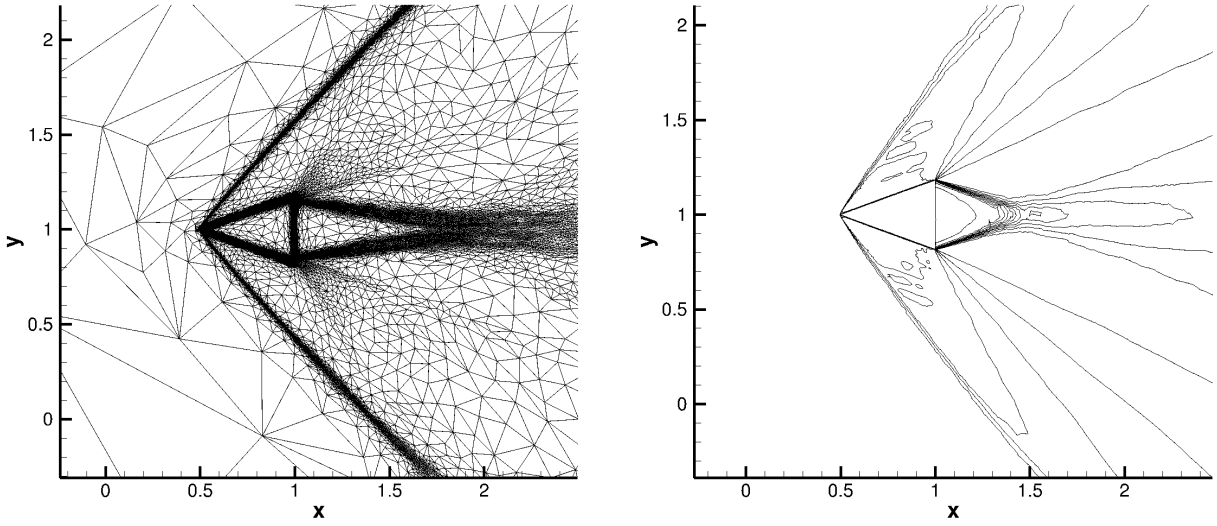


Figure 3.8: Final adapted mesh and solution (Mach number) for test case 3.5.2.

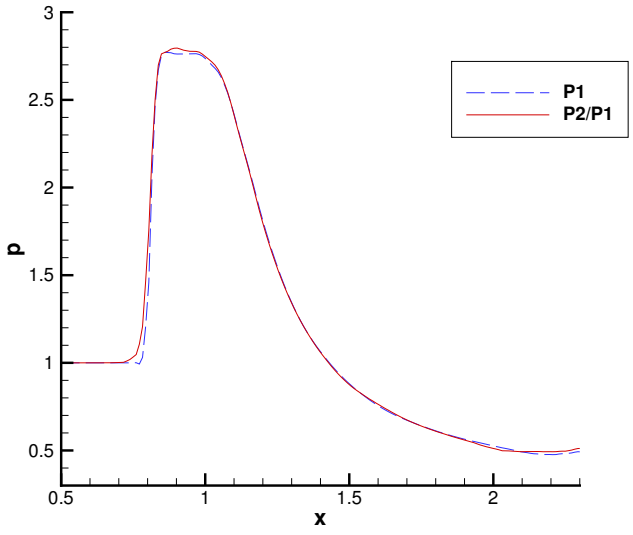


Figure 3.9: Comparison of pressure on line $y = 1.44$ with p-adaptation and $\mathbb{P}^1$ elements.

3.5.3 Three dimensional supersonic prism

For this final test case, we study the three dimensional flow field over a prism moving at a supersonic speed. The height of the prism is $h = 1$, its width is $w = 0.5$ and its length is $l = 2$. The prism is placed inside a rectangular parallelepiped of length $l = 6.5$ and width $w = 4$ (see figure 3.10). On the boundary 1 of the rectangular parallelepiped we impose the Steger-Warming boundary conditions with the values $(\rho, u, v, w, p) = (1, 2.366, 0, 0, 1)$ and on the boundary 2 we impose the no-slip boundary condition $\mathsf{u} \cdot \mathbf{n} = 0$.
The variables are initialized with $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 2.5, 0, 1)$ and the Reynolds number is set at $Re_\infty = 50000$. The mesh is again adapted with three cycles of adaptation, with the following parameters for both physical and level-set adaptation:

- $\epsilon = 0.005$,
- $h_{\min} = 0.005$,
- $h_{\max} = 0.5$.

We have used greater values than those used for the test cases of dimension two, in order to reduce the mesh sizes and lower the computational time.

As in the two dimensional test case, we use the shock detector (3.44). The final mesh is then constituted of quadratic elements (which are now Bézier elements (3.23) for the reasons explained in section 3.4.2) except in the penalized and shock zone, where $\mathbb{P}^1$ elements are used, as shown in figure 3.11. The solution (x-component of the velocity) is shown in figure 3.12. The solution looks physically admissible, to the best of our knowledge.

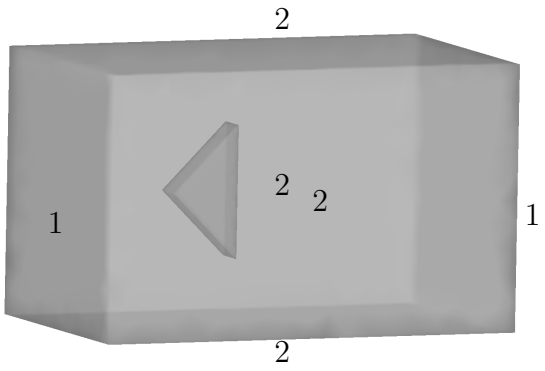


Figure 3.10: Boundary conditions for test case 3.5.3.

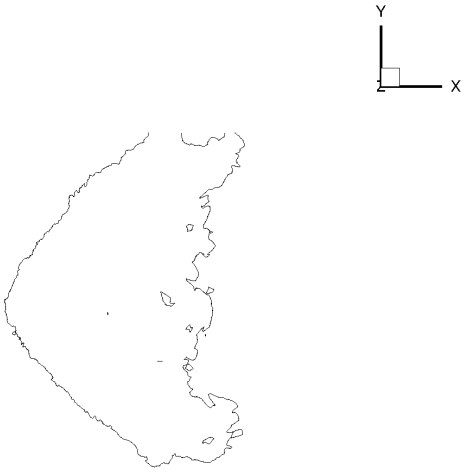


Figure 3.11: Contour of zone with $\mathbb{P}^1$ elements for test case 3.5.3.

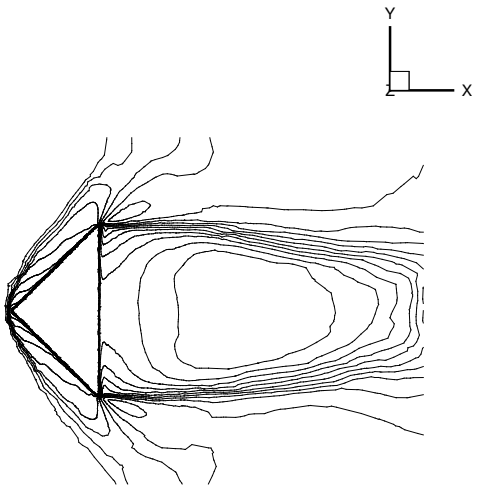


Figure 3.12: Isolines of Mach number for test case 3.5.3 (two dimensional slice cut at the middle of the prism in the z direction).

3.5.4 Conclusion

We have presented in this work an hp-adaptive Residual Distribution scheme applied to the resolution of the penalized Navier-Stokes equations. The resulting method constitutes an evolution of the IBM-LS-AUM method in the sense that it is now possible to use it with high order elements within a Residual Distribution scheme. We have presented numerical simulations that show the robustness of the method for subsonic, transonic, supersonic and hypersonic test cases. The method extends to three dimensional cases, as shown with the complex test case of a supersonic delta wing.
From the recent results about p-adaptation for Residual Distribution scheme cited in this work, it will be possible to extend the method to finite elements of higher order (cubic and beyond). This may be the subject of a future publication. A longer term project could be the study of the hp-adaptive Residual Distribution scheme for time dependent equations.

Acknowledgments

- This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02).

- Experiments presented in this chapter were carried out using the cluster Avakas and the PLAFRIM experimental testbed. PLAFRIM is developed under the Inria PlaFRIM development action with support from LABRI, IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS.

Conclusion

Summary

In this study, we have presented the construction of a p-adaptive Residual Distribution scheme for the Euler equations and an hp-adaptive Residual Distribution scheme for the penalized Navier-Stokes equations. The construction of the p-adaptive Residual distribution scheme is based on formulas that express the total residual of an element of arbitrary polynomial degree in terms of the combination of the total residuals of the sub-elements. The discrete solution obtained with the p-adaptive scheme is not continuous across the interface of a subdivided element and a non subdivided element. This contradicts the continuity constrain imposed to the solution of a continuous Residual Distribution scheme. However, we have shown that with a suitable combination of quadrature formulas this continuity constrain can be alleviated and the Lax-Wendroff theorem can still be applied to the p-adaptive scheme, thus ensuring its convergence to the correct weak solution. Different test cases at various speed regimes in dimension two and three have demonstrated the robustness of the method. The p-adaptive scheme has then been applied to the penalized Navier-Stokes equations. Coupled with an exterior anisotropic mesh adaptation method, we have obtained an extension of the IBM-LS-AUM algorithm to high order elements in the frame of a Residual Distribution scheme. The numerical results have demonstrated the robustness of the method in dimension two and three at various speed regimes and different Reynolds numbers. Finally, the methods presented in this work are by designed by construction for the Residual Distribution schemes and so they do not change fundamentally the Residual Distribution method. As a consequence, the p-adaptation mechanism has been integrated without profound modifications to the original code (the work has been more intensive on the theoretical aspects) and the implementation of the penalization method was done with no particular problems. A summary of the principal aspects of the numerical implementation within the platform RealFluid can be found in Appendix A.

Possible future extensions of this work

Higher order polynomials
The method presented in this work can be theoretically extended to any polynomial order, as shown by the formulas of chapter 2. However, only polynomials of degree two have been considered for the numerical experiments. It could then be interesting to experiments with higher degree polynomials. However, this would need a careful implementation, notably of the parallelization of the code, in order to clearly see the advantages in terms of speed brought by the use of higher order polynomials for a given level of accuracy.

Other type of finite elements
In the work presented here, only simplices have been envisaged. We could try to extend the results obtained in this work to other type of finite elements, like cubic or prismatic finite elements. The subdivision formulas have been studied only for triangles and tetrahedrons and it would be interesting to further generalize them to different type of finite elements. Moreover, this could lead to the construction of a p-adaptive Residual Distribution scheme able to handle hybrid meshes.

Improvement of the mesh adaptation method
In the implementation used in this work, the anisotropic mesh adaptation is external to the code and it could be suitable to integrate it into the code, so that the hp-adaptation process is fully automated. In the work presented here, only feature-based hp-adaption has been considered. It could be interesting to study the implementation of an adjoint-based (also called output-based) adaptation method in the frame of Residual Distribution schemes. The adjoint-based mesh adaptation method used in Finite Element methods, Finite Volume methods and Discontinuous Galerkin methods consists in adapting the mesh until a desired level of accuracy is achieved for a particular output variable (like for example the lift and drag). However the method seems to be computationally expensive and its integration into a Residual Distribution framework does not seem easy.

Extension to unsteady problems
Finally, an ambitious objective would be to extend the work presented here to unsteady problems. However, it should be pointed out that the work presented here, that is hp-adaptation, is the natural step after high order Residual Distribution schemes have been designed. As the construction of high order unsteady Residual Distribution schemes has not yet been achieved, the construction of a hp-adaptive unsteady Residual Distribution scheme is more a long term goal. Moreover, the p-adaptive scheme presented in this thesis relies on the formulas of chapter 2 and so they need to be extended to the unsteady

case.

Appendix A

The computational code: a summary of the practical implementation of the p-adaptation method

We present here a rapid summary of the code implementation of the p-adaptation method in the platform Real Fluid.
The code implementation of the algorithm of the penalization method presented in chapter 3 has been done with no major difficulties and easily combines with the implementation of the p-adaption method, so we do not detail it here.

We consider for simplicity the two dimensional case with $\mathbb{P}^2$ and $\mathbb{P}^1$ triangles (unless stated otherwise). The extension to $\mathbb{P}^k$ elements in dimension two and three is straightforward.

Construction of sub-elements
Once the mesh has been read, the $\mathbb{P}^2$ triangles are constructed. For each $\mathbb{P}^2$ triangle, four sub-triangles are constructed. To do so, a constructor of sub-triangles is added to the type triangle and is called at the construction of the triangles. Similarly, for the boundary elements (which are segments here), a constructor of sub-segments is added to the type segment and is called at the construction of segments.

Time loop
Once the elements are constructed, the equation (2.21) is solved iteratively with a time loop (67)-(68). For each time iteration, we first perform an iterative loop over the triangles of the mesh and then, in the case of weak boundary conditions, an iterative loop over the boundary segments of the mesh is performed.

Iterative loop over the domain triangles

If the triangle K is not subdivided, its total residual Φ^K and its nodal residuals $\Phi^K_\sigma, \sigma \in \Sigma_K$ are computed. For a given node σ of K, the nodal residual Φ^K_σ is added to the global sum of all the contributions received by this node coming from the elements containing the node σ, in order to contribute to equation (2.21).

If the triangle K is subdivided, a loop over its sub-triangles is performed and for each sub-triangle ξ the same operations as above are performed. The difference is that each nodal residual Φ^ξ_σ is multiplied by the subdivision coefficient γ_ξ corresponding to the sub-triangle ξ in which it is computed, as explained in section 2.3.5.

Iterative loop over the boundary segments

The procedure for boundary elements is the same as for domain elements. If the segment Γ is not subdivided, the boundary nodal residuals $\Phi^\Gamma_\sigma, \sigma \in \Sigma_\Gamma$ are computed and added to the global sum of all contributions received by the node σ, in order to contribute to equation (2.21).

If the segment Γ is subdivided, a loop over its sub-segments is performed and for each sub-segment ς the same operations are performed, with the difference that, like for a domain element, the nodal residuals $\Phi^\varsigma_\sigma, \sigma \in \Sigma_\varsigma$ are multiplied by the subdivision coefficient γ_ξ corresponding to the sub-triangle ξ containing the sub-segment ς, as explained in section 2.3.5.

Computation of the Jacobian matrix

In the case of an implicit scheme, we need to compute the Jacobian matrix J (66), which is assembled for each node σ of the mesh from the sum of all the contributions brought by the elements containing σ.

When the domain triangle K is subdivided, the contribution brought to J by the node σ in the sub-element ξ of K is multiplied by the subdivision coefficient γ_ξ corresponding to the sub-triangle ξ, as explained in section 2.3.5.

For a subdivided boundary segment Γ, in the same manner the contribution to J brought by the node σ in the sub-element ς is multiplied by γ_ξ, the subdivision coefficient of the sub-triangle containing the sub-segment ς.

Gradient reconstruction

At the beginning of each time iteration, the gradient of the solution is reconstructed with exactly the same method as in [9]. Consequently, the p-adaptation method presented in this work has no consequence on the method used to reconstruct the gradient of the solution. However, we note that the reconstructed gradient $\nabla \mathbf{u}_h$ is in general not continuous at the interface between a subdivided triangle and a non subdivided triangle like the solution $\mathbf{u}_h$, but, with the same arguments as those developed for the solution $\mathbf{u}_h$, this is not a

problem for the application of the Lax-Wendroff theorem.

Implementation of quadrature formulas

The equality (2.40) and the formulas presented in the section 2.3 determine the choice of the quadrature formulas as we have seen. We precise now their implementation in the computer code. We use a numerical implementation that satisfies numerically the relation 2.40, which we recall here

$$\int_{\Gamma} \varphi_\sigma \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} \quad + \int_{\Gamma'} \varphi_\sigma \mathbf{f}(\mathbf{u}_h) \cdot \mathbf{n} = 0. \tag{A.1}$$

To do so, we use quadrature formulas such that the quadrature points are the same for all the elements of the mesh.

The formulas of chapter 2 are given for $\mathbf{f}^h(\mathbf{u}_h)$, however another natural choice in the implementation could be to use $\mathbf{f}(\mathbf{u}_h)$. These quantities are not the same, indeed $\mathbf{f}(\mathbf{u}_h)$ is the true flux applied to the interpolant of the solution:

$$\mathbf{f}(\mathbf{u}_h) = \mathbf{f}(\sum_{\sigma \in \Sigma_K} \mathbf{u}_i \varphi_i), \tag{A.2}$$

while $\mathbf{f}^h(\mathbf{u}_h)$ is the interpolant of the flux:

$$\mathbf{f}^h(\mathbf{u}_h) = \sum_{\sigma \in \Sigma_K} \mathbf{f}(\mathbf{u}_i) \varphi_i. \tag{A.3}$$

As showed in chapter 2, if we write the subdivision formula in its generic form, we have the following equality

$$\sum_{T \in \mathscr{T}_{h_S}} \sum_{\xi \in T_S} \gamma_\xi \int_{\partial \xi} \mathbf{f}^h(\mathbf{u}_h) \cdot \mathbf{n} \, dx = \sum_{T \in \mathscr{T}_h} \int_{\partial T} \mathbf{f}^h(\mathbf{u}_h) \cdot \mathbf{n} \, dx \tag{A.4}$$

where $\mathscr{T}_{h_S}$ denotes the set of subdivided triangles of $\mathscr{T}_h$, T_S denotes the set of sub-triangles of a given triangle T and $\mathbf{f}^h(\mathbf{u}_h)$ represents the interpolant of the flux (as defined above) local to the element (meaning that it is a polynomial of degree 1 if it is computed in a divided element or a polynomial of degree k in a non divided element).

Now, if we replace $\mathbf{f}^h(\mathbf{u}_h)$ by $\mathbf{f}(\mathbf{u}_h)$, then the equality (A.4) is not verified in general, because the subdivision formula is not exact anymore.

A natural solution is then to use quadrature formulas such that the quadrature points are equal to the nodes of the elements, because then, with the symbol $\oint$ designating such a quadrature formula, $\oint \mathbf{f}(\mathbf{u}_h)$ reduces to $\oint \mathbf{f}^h(\mathbf{u}_h)$ and so the relation (A.4) is verified. This is the solution that has been retained in this work in the case of dimension two.

However, in dimension three with quadratic tetrahedral elements (so with triangular faces), if we use such a quadrature formula, the weights corresponding

to the vertices of the triangles are necessarily equal to zero.
Indeed, if ψ is a polynomial quadratic function defined on the triangle K, it is uniquely decomposed on the Lagrange basis of $\mathbb{P}^2(K)$ as

$$\psi = \sum_{i=1}^{6} \alpha_i \varphi_i \ . \tag{A.5}$$

So, we can exactly integrate ψ with the following quadrature formula

$$\int_K \psi = \sum_{i=1}^{6} \alpha_i \int_K \varphi_i \ . \tag{A.6}$$

As we have

$$\alpha_j = \sum_{i=1}^{6} \alpha_i \delta_{ij} = \sum_{i=1}^{6} \alpha_i \varphi_i(a_j) = \psi(a_j) \ , \tag{A.7}$$

the quadrature formula writes

$$\int_K \psi = \sum_{i=1}^{6} \psi(a_i) \int_K \varphi_i \ . \tag{A.8}$$

It is unique, indeed if we write

$$\int_K \psi = \sum_{i=1}^{6} \psi(a_i) \omega_i \ , \tag{A.9}$$

we have

$$\omega_j = \sum_{i=1}^{6} \omega_i \delta_{ij} = \sum_{i=1}^{6} \omega_i \varphi_j(a_i) = \int_K \varphi_j \ , \tag{A.10}$$

where the last equality comes from relation (A.9).

In our implementation with this quadrature formula, the numerical experiments tended to fail, probably because not enough points contribute to the total residual.
The retained solution is then to use the interpolant of the flux $\mathbf{f}^h(\mathbf{u}_h)$ but with quadrature formulas of higher orders, and such that the quadrature points coincide everywhere.

Implementation of generic subdivision formulas
To implement the generic subdivision formulas based on the Bézier approximation of the flux $\mathbf{f}^B$, we have to compute, for example in the case of the two-dimensional formula (2.48), the following integrals:

$$\int_{K_{(i-1)^2+1}} \text{div } \tilde{\mathbf{f}}^{(1)} dx, \int_{K_{(i-1)^2+3}} \text{div } \tilde{\mathbf{f}}^{(1)} dx, ..., \int_{K_{i^2-2}} \text{div } \tilde{\mathbf{f}}^{(1)} dx, \int_{K_{i^2}} \text{div } \tilde{\mathbf{f}}^{(1)} dx \tag{A.11}$$

for $i = 1, ..., k$.

For each integral, we have to compute the quantity (2.50) which we recall here

$$\tilde{\mathbf{f}}^{(1)} = \mathbf{f}_{l_1}\lambda_1 + \mathbf{f}_{l_2}\lambda_2 + \mathbf{f}_{l_3}\lambda_3 \ , \tag{A.12}$$

where the functions $\lambda_1, \lambda_2, \lambda_3$ are the $\mathbf{P}^1$ basis functions of the big element K (*not* of the sub-element). To do so, we have to implement in the code for each sub-element a way to access (for example a pointer) the $\mathbf{P}^1$ basis functions of the big element (and so they need to be constructed).
To compute the quantities $\mathbf{f}_{l_1}, \mathbf{f}_{l_2}$and $\mathbf{f}_{l_3}$ we make the following observation: we remark that the quadratic Bézier approximation $\mathbf{f}^B$ as defined by (2.42) is in fact equal to the Lagrange interpolation denoted here by $\mathbf{f}^L$ (and hence the Bézier approximation is an interpolation too). Indeed, the Bézier expansion of a function ψ writes

$$\psi = \sum_{\sigma \in \Sigma_K} \psi_\sigma B_\sigma \tag{A.13}$$

where ψ_σ is the value of ψ at the Bézier control point σ. So, with $\mathbf{f}^L$ the Lagrange interpolant of the function $\mathbf{f}$ on the element K, the Bézier expansion of $\mathbf{f}^L$ writes

$$\mathbf{f}^L = \sum_{\sigma \in \Sigma_K} \mathbf{f}^L_\sigma B_\sigma \tag{A.14}$$

where $\mathbf{f}^L_\sigma$ is the value of $\mathbf{f}^L$ at the Bézier control point σ, which is equal to $\mathbf{f}^L(\sigma)$ if σ is a vertex of K and defined by

$$\mathbf{f}^L_\sigma = 2\mathbf{f}^L(\sigma) - \frac{\mathbf{f}^L(\sigma_1) + \mathbf{f}^L(\sigma_2)}{2} \tag{A.15}$$

if σ is a non vertex node of K (σ_1 and σ_2 are the vertices of the segment containing the node σ). As $\mathbf{f}^L$ is by definition the interpolant of $\mathbf{f}$ on K, we have for a non vertex node

$$\mathbf{f}^L_\sigma = 2\mathbf{f}(\sigma) - \frac{\mathbf{f}(\sigma_1) + \mathbf{f}(\sigma_2)}{2} = \mathbf{f}_\sigma \tag{A.16}$$

and so we have $\mathbf{f}^L_\sigma = \mathbf{f}_\sigma, \forall \sigma \in \Sigma_K$ and finally

$$\mathbf{f}^L = \sum_{\sigma \in \Sigma_K} \mathbf{f}^L_\sigma B_\sigma = \sum_{\sigma \in \Sigma_K} \mathbf{f}_\sigma B_\sigma = \mathbf{f}^B \ . \tag{A.17}$$

From this observation we define, for an arbitrary polynomial degree of approximation k, the values $\mathbf{f}_\sigma$ of $\mathbf{f}$ at the control points so that the Bézier approximation $\mathbf{f}^B$ is equal to the Lagrange interpolation $\mathbf{f}^L$ as in the quadratic case.
We can now compute the quantities $\mathbf{f}_{l_i}$. For this purpose, we introduce the matrix $M = M(i,j), i, j = 1, ..., N_h$ (we recall that N_h is the cardinal of Σ_h)

such that $M(i,j) = B_j(a_i)$ where $B_j(a_i)$ is the Bézier basis function associated to the node a_j evaluated at the node a_i, with $a_i, a_j \in \Sigma_h$, and we solve the linear system

$$M \begin{pmatrix} \mathbf{f}_{a_1} \\ \vdots \\ \mathbf{f}_{a_{N_h}} \end{pmatrix} = \begin{pmatrix} \mathbf{f}(a_1) \\ \vdots \\ \mathbf{f}(a_{N_h}) \end{pmatrix} . \tag{A.18}$$

Bibliography

[1] R. Abgrall. Toward the ultimate conservative scheme: following the quest. *Journal of Computational Physics*, 167(2):277–315, 2001.

[2] R. Abgrall. Toward the ultimate conservative scheme: following the quest. *Journal of Computational Physics*, 167(2):277–315, 2001.

[3] R. Abgrall. Essentially non-oscillatory Residual Distribution schemes for hyperbolic problems. *Journal of Computational Physics*, 214(2):773–808, 2006.

[4] R. Abgrall. A Residual Distribution method using discontinuous elements for the computation of possibly non smooth flows. *Advances in Applied Mathematics and Mechanics*, 2(1):32–44, 2010.

[5] R. Abgrall. A review of Residual Distribution schemes for hyperbolic and parabolic problems: the July 2010 state of the art. *Communications in Computational Physics*, 11(04):1043–1080, 2012.

[6] R. Abgrall and T. Barth. Residual Distribution schemes for conservation laws via adaptive quadrature. *SIAM Journal on Scientific Computing*, 24(3):732–769, 2003.

[7] R. Abgrall and T. Barth. Residual distribution schemes for conservation laws via adaptive quadrature. *SIAM Journal on Scientific Computing*, 24(3):732–769, 2003.

[8] R. Abgrall, H. Beaugendre, and C. Dobrzynski. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257:83–101, 2014.

[9] R. Abgrall and D. De Santis. Linear and non-linear high order accurate Residual Distribution schemes for the discretization of the steady compressible Navier-Stokes equations. *Journal of Computational Physics*, 283:329–359, 2015.

[10] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of very high order Residual Distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *Journal of Computational Physics*, 230(11):4103–4136, 2011.

[11] R. Abgrall, A. Larat, M. Ricchiuto, and C. Tavé. A simple construction of very high order non-oscillatory compact schemes on unstructured meshes. *Computers and Fluids*, 38(7):1314–1323, 2009.

[12] R. Abgrall and P. L. Roe. High order fluctuation schemes on triangular meshes. *Journal of Scientific Computing*, 19(1-3):3–36, 2003.

[13] R. Abgrall and P. L. Roe. High order fluctuation schemes on triangular meshes. *Journal of Scientific Computing*, 19(1-3):3–36, 2003.

[14] R. Abgrall and J. Trefilik. An example of high order Residual Distribution scheme using non-lagrange elements. *Journal of Scientific Computing*, 45(1-3):3–25, 2010.

[15] R. Abgrall and J. Trefilik. An example of high order Residual Distribution scheme using non-lagrange elements. *Journal of Scientific Computing*, 45(1-3):3–25, 2010.

[16] R. Abgrall, Q. Viville, H. Beaugendre, and C. Dobrzynski. p-adaptation using Residual Distribution schemes with continuous finite elements. Research Report RR-8808, Inria Bordeaux Sud-Ouest ; IMB ; Bordeaux INP, November 2015.

[17] B. R. Ahrabi. An hp-adaptive Petrov-Galerkin method for steady-state and unsteady flow problems. *PhD, Department of Computational Engineering, University of Tennesee at Chattanooga, Chattanooga, TN*, 2015.

[18] B. R. Ahrabi, W. K. Anderson, and J. C. Newman. High-order finite-element method and dynamic adaptation for two-dimensional laminar and turbulent Navier-Stokes. AIAA paper 2014-2983, 2014. 32nd AIAA Applied Aerodynamics Conference.

[19] B. R. Ahrabi, W. K. Anderson, and J.C. Newman. An adjoint-based hp-adaptive Petrov-Galerkin method for turbulent flows. AIAA 2015-2603, 2015. 22nd AIAA Computational Fluid Dynamics Conference.

[20] J. D. Anderson. Computational fluid dynamics: the basics with applications. 1995. *McGrawhill Inc.*

[21] P. Angot, C. H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, 1999.

[22] I. Babuška and M. Suri. The p-and hp versions of the finite element method, an overview. *Computer Methods in Applied Mechanics and Engineering*, 80(1-3):5–26, 1990.

[23] I. Babuška and B.Q. Guo. The h, p and h-p version of the finite element method; basis theory and applications. *Adv. Eng. Softw.*, 15(3-4):159–174, 1992.

[24] H. Beaugendre, L. Nouveau, C. Dobrzynski, R. Abgrall, and M. Ricchiuto. Unsteady residual distribution schemes adapted to immersed boundary methods on unstructured grids to account for moving bodies. In *13th US National Congress on Computational Mechanics*, 07 2015.

[25] C. Bernardi, Y. Maday, and F. Rapetti. *Discrétisations variationnelles de problèmes aux limites elliptiques*, volume 45. Springer Science & Business Media, 2004.

[26] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large mach number flows in the presence of obstacles. *Computers and fluids*, 38(3):703–714, 2009.

[27] A. N. Brooks and T. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.

[28] A. Burbeau, P. Sagaut, and C.-H. Bruneau. A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 169(1):111–150, 2001.

[29] A. Cangiani, J. Chapman, E. H. Georgoulis, and M. Jensen. Implementation of the continuous-discontinuous Galerkin finite element method. *Numerical Mathematics and Advanced Applications 2011*, page 315.

[30] J. Cheng and C. W. Shu. High order schemes for cfd: A review. *Chinese Journal of Computational Physics*, 5:002, 2009.

[31] B. Cockburn, S.-Y. Lin, and C. W. Shu. Tvb Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws iii: one-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989.

[32] C. Dapogny and P. Frey. Computation of the signed distance function to a discrete contour on adapted triangulation. *Calcolo*, 49(3):193–219, 2012.

[33] E. F. D'Azevedo and R. B. Simpson. On optimal triangular meshes for minimizing the gradient error. *Numerische Mathematik*, 59(1):321–348, 1991.

[34] D. De Santis. *Development of a high-order Residual Distribution method for Navier-Stokes and RANS equations*. PhD thesis, Université Sciences et Technologies-Bordeaux I, 2013.

[35] H. Deconinck, H. Paillere, R. Struijs, and P. L. Roe. Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. *Computational Mechanics*, 11(5-6):323–340, 1993.

[36] H. Deconinck, R. Struijs, G. Bourgois, and P. L. Roe. Compact advection schemes on unstructured grids. Computational Fluid Dynamics, VKI LS 1993-04, 1993. SEE N94-18557 04-34.

[37] L. Demkowicz. *Computing with hp-ADAPTIVE FINITE ELEMENTS: Volume 1 One and Two Dimensional Elliptic and Maxwell problems*. CRC Press, 2006.

[38] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69. Springer Science & Business Media, 2011.

[39] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003.

[40] V. Ducrot and P. J. Frey. Contrôle de l'approximation géométrique d'une interface par une métrique anisotrope. *Comptes Rendus Mathématiques*, 345(9):537–542, 2007.

[41] J. A. Ekaterinaris. High-order accurate, low numerical diffusion methods for aerodynamics. *Progress in Aerospace Sciences*, 41(3):192–300, 2005.

[42] J. T. Erwin, W. K. Anderson, S. Kapadia, and L. Wang. Three-dimensional stabilized finite elements for compressible Navier-Stokes. *AIAA Journal*, 51(6):1404–1419, 2013.

[43] J T. Erwin, W. K. Anderson, L. Wang, and S. Kapadia. High-order finite-element method for three-dimensional turbulent Navier-Stokes. AIAA 2013-2571, 2013. 21st AIAA Computational Fluid Dynamics Conference.

[44] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[45] R. P. Fedkiw. Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *Journal of Computational Physics*, 175(1):200–224, 2002.

[46] L. P. Franca and S. L. Frey. Stabilized finite element methods: Ii. the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3):209–233, 1992.

[47] P. J. Frey and F. Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48):5068–5082, 2005.

[48] P. J. Frey and P. L. George. Mesh generation: application to finite elements, hermes science. *Europe*, 2000.

[49] G. Gassner, F. Lörcher, and C. D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 224(2):1049–1063, 2007.

[50] R. Ghias, R. Mittal, and H. Dong. A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics*, 225(1):528–553, 2007.

[51] A. Gilmanov and F. Sotiropoulos. A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457–492, 2005.

[52] A. Guardone, D. Isola, and G. Quaranta. Arbitrary Lagrangian Eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping. *Journal of Computational Physics*, 230(20):7706–7722, 2011.

[53] R. Hartmann. Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 51(9-10):1131–1156, 2006.

[54] R. Hartmann. Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity. *Lecture Series - Von Karman Institute for Fluid Dynamics*, 1:5, 2006.

[55] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.

[56] X. Y. Hu, B. C. Khoo, N. A. Adams, and F. L. Huang. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219(2):553–578, 2006.

[57] T. Hughes, G. Scovazzi, and T. E. Tezduyar. Stabilized methods for compressible flows. *Journal of Scientific Computing*, 43(3):343–368, 2010.

[58] T. Hughes and T. E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45(1):217–284, 1984.

[59] G. E. Karniadakis, C. W. Shu, and B. Cockburn. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, 2000.

[60] S. M. Kast and K. J. Fidkowski. Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains. *Journal of Computational Physics*, 252:468–494, 2013.

[61] B. Kirk and T. Oliver. Validation of SUPG finite element simulations of Shockwave/Turbulent boundary layer interactions in hypersonic flows. AIAA 2013-306, 2013. 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, Grapevine, TX, USA.

[62] B. S. Kirk and G. F. Carey. Development and validation of a SUPG finite element scheme for the compressible Navier-Stokes equations using a modified inviscid flux discretization. *International Journal for Numerical Methods in Fluids*, 57(3):265–293, 2008.

[63] A. Larat. *Conception et analyse de schémas distribuant le résidu d'ordre très élevé. Application à la mécanique des fluides*. PhD thesis, Ph. D. thesis, Université Sciences et Technologies-Bordeaux I, 2009.

[64] C. Lepage and W. Habashi. Fluid-structure interactions using the ALE formulation. AIAA paper 1999-0660, 1999. 37th Aerospace Sciences Meeting and Exhibit.

[65] C. Lepage and W. Habashi. Conservative interpolation of aerodynamic loads for aeroelastic computations. AIAA 2000-1449, 2000. 41st Structures, Structural Dynamics, and Materials Conference and Exhibit.

[66] R. Löhner, J. D. Baum, E. Mestreau, D. Sharov, C. Charman, and D. Pelessone. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering*, 60(3):641–660, 2004.

[67] Andreas Mark and B. G. M. Van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680, 2008.

[68] K. Masatsuka. *I Do Like CFD, Vol. 1*, volume 1. Lulu.com, 2013.

[69] W. F. Mitchell and M. A. McClain. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):2, 2014.

[70] W. F. Mitchell and M. A. McClain. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):2, 2014.

[71] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[72] R. Mittal, V. Seshadri, and H. S. Udaykumar. Flutter, tumble and vortex induced autorotation. *Theoretical and Computational Fluid Dynamics*, 17(3):165–170, 2004.

[73] D. Moxey, M. D. Green, S. J. Sherwin, and J. Peiró. An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering*, 283:636–650, 2015.

[74] R.-H. Ni. A multiple-grid scheme for solving the euler equations. *AIAA journal*, 20(11):1565–1571, 1982.

[75] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[76] J. F. Remacle, J. E. Flaherty, and M. S. Shephard. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review*, 45(1):53–72, 2003.

[77] J. F. Remacle, X. Li, M. S. Shephard, and J. E. Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, 62(7):899–923, 2005.

[78] M. Ricchiuto. *Contributions to the development of residual discretizations for hyperbolic conservation laws with application to shallow water flows.* Habilitation à diriger des recherches, Université Sciences et Technologies - Bordeaux I, 2011.

[79] K. Shahbazi, P. F. Fischer, and C. R. Ethier. A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations. *Journal of Computational Physics*, 222(1):391–407, 2007.

[80] C.-W. Shu. High order ENO and WENO schemes for computational fluid dynamics. In *High-order methods for computational physics*, volume 9 of *T.J. Barth and H. Deconinck, editor, Lecture Notes in Computational Science and Engineering*, pages 439–582. Springer, 1999.

[81] P. Solin, K. Segeth, and I. Dolezel. *Higher-order finite element methods.* CRC Press, 2003.

[82] S. Tan, C. Wang, C. W. Shu, and J. Ning. Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. *Journal of Computational Physics*, 231(6):2510–2527, 2012.

[83] K. Wang, A. Rallu, J. F. Gerbeau, and C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 67(9):1175–1206, 2011.

[84] L. Wang. *Techniques for high-order adaptive discontinuous Galerkin discretizations in fluid dynamics.* PhD thesis, 2009.

[85] Z. Wang. A perspective on high-order methods in computational fluid dynamics. *Science China Physics, Mechanics & Astronomy*, 59(1):1–6, 2016.

[86] Z. J. Wang. High-order methods for the Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43(1):1–41, 2007.

[87] Z. J. Wang. High-order computational fluid dynamics tools for aircraft design. *Philosophical Transactions. Series A, Mathematical, physical, and engineering sciences*, 372(2022), 2014.

[88] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[89] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.

[90] M. Woopen, G. May, and J. Schütz. Adjoint-based error estimation and mesh adaptation for hybridized discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 76(11):811–834, 2014.

[91] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering*, 24(2):337–357, 1987.

[92] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.